MODEL 170
Q-MONITOR
USER'S MANUAL

CODAR
TECHNOLOGY
INC.

BLANK

MODEL 170
Q-MONITOR
USER'S MANUAL

Publication No. 130-0006 REV A
Printed in USA 10/87
Copyright 1985 CodarTechnology,Inc.

Codar Technology Inc.
Computer Products Division
1500 Kansas Ave., Bldg. 2E
Longmont, Colorado 80501 U.S.A.
Telex No. 4947090 Codar
(303) 776-0472

BLANK

NOTE ON MANUAL

THIS MANUAL IS ORGANIZED INTO TWO SEPERATE SECTIONS:

PART I DEALS MOSTLY WITH THE HARDWARE

PART II DEALS MOSTLY WITH THE SOFTWARE

EACH SECTION HAS A SEPERATE TABLE OF CONTENTS,
AND A SEPERATE SET OF APPENDICES (INCLUDES INDEX)

PART II BEGINS AFTER THE COLORED PAPER DIVIDER

BLANK

13-May-87

MODEL 170 Q-MONITOR

INSTALLATION GUIDE

BLANK

INSTALLATION INDEX

A/D converter 4, 10
address PAL 2, 3, 12

battery 2, 10, 13
baud rate 5, 12
BEVNT 3, 12
BREAK 6, 12

C-D backplane 2, 5
calendar clock 10
character format 5
console terminal 3, 7, 12, 13

digital I/O port 3, 4, 9

Failover 4

handling 2, 10

interrupt priorities 6, 12
interrupt vectors 7, 12
interrupts 5, 7, 8

line time clock 3, 8, 12

MicroVAX 2, 7

oscillator 10

parity 5, 12

Q-Bus address 3, 7
Q-Bus compatibility 2, 5
QMinit 6, 13
QMsetup 2, 5, 6, 7, 8, 11, 12

serial I/O 5, 7, 8

voltage reference 10

BLANK

This manual will guide you through installation of a Q-Monitor in your computer system. There are a number of things to be done before using the Q-Monitor, falling into three categories:

1) Hardware configuration (address PAL and jumpers) that should be done prior to installing the board in your system's backplane;

2) Firmware configuration (SET and SAVE commands) that should be done after installing the board, but prior to bootstrapping your operating system; and

3) Firmware configuration that can be done while your system is running.

Most of the Q-Monitor's characteristics fall into the last category -- there are only a few features that must be configured during installation. Features which need not be configured during installation are generally not covered in this manual (see "Q-Monitor User's Guide" and "Q-Monitor Reference Manual").

Q-Monitor firmware can be configured in either of two ways: by using the on-board command processor known as "QMsetup", or by issuing escape sequences from a host-resident program. For the purposes of installation, QMsetup is preferred and is the only technique disussed in this manual.

NOTE

The Q-Monitor is compatible with all 16-, 18-, and 22-bit Q-Bus systems; on a MicroVAX (or any system using a "C-D backplane") care must be taken not to plug the Q-Monitor into a C-D slot.

Refer to Appendix A of this manual for component locations.

When handling the Q-Monitor, avoid touching the calendar clock and oscillator circuitry near U31 and Y3.

The Q-Monitor is shipped with a nonconductive plastic sheet on the solder side of the board; this sheet protects the Lithium battery and associated circuitry from shorting should the board be placed on a conductive surface. If the plastic sheet is removed, it must be reattached with noncorrosive RTV adhesive. Corrosive RTV can usually be identified by a strong vinegar-like acetic acid smell.

1.0 Before you plug it in ...

1.1 Q-Bus addressing

The Q-Monitor has four registers which require four consecutive 16-bit word locations in the Q-Bus I/O page; the set of addresses at which these registers appear is controlled by the PAL at U14. The normal (non-custom) options are:

LSI-11 Console	LSI-11 Non-console		MicroVAX
17 777560	17 775610	receiver control and status (RCSR)	2000FB88
17 777562	17 775612	receiver data buffer (RBUF)	2000FB8A
17 777564	17 775614	tranmitter control and status (TCSR)	2000FB8C
17 777566	17 775616	transmitter data buffer (TBUF)	2000FB8E

(If your processor is an LSI 11/73, you must type the extra "17" when using console ODT; if it's a MicroVAX, the console terminal address does not apply. Other addresses configurations can be arranged by special order.)

If you specified the Q-bus address when ordering your Q-Monitor, verify that the address label on the PAL at U14 matches the address you specified; or, if you received more than one address PAL, install the one you wish to use (making sure that the notched end of the PAL is aligned with the notched end of the socket).

Wherever your Q-Monitor is to appear in the I/O page, you must verify that no other device on the bus uses any of the same four addresses. You can do this (PRIOR to installing the board) using your host processor's console ODT facility: when you attempt to open each of the four locations you should get a "non-existent memory" response. If not, you must determine the source of the address conflict and correct it (by changing the addressing of either device) BEFORE installing the Q-Monitor.

1.2 Line Time Clock (LTC)

As shipped, one of the Q-Monitor's counter/timers (CT1) and one of its digital I/O port lines (bit <4>) are configured to supply a 50 Hz square wave output, which can be used to drive BEVNT as a 50 Hz LTC; however, the standard jumper configuration (E18-E19, E21-E22) is such that this signal is not actually connected to BEVNT on the Q-Bus (since most systems will already have an LTC). If you will not be using the Q-Monitor as the system LTC, leave these jumpers as they are.

If the Q-Monitor is to provide the system LTC, remove the jumper at pins E18-E19 and insert it at E17-E18; also, remove the jumper at E21-E22 and insert it at E20-E21. You must also verify that no other devices drive the BEVNT line. Multiple BEVNTs will produce an erratic LTC at best; more likely they will prevent your system from booting properly.

1.3 Digital I/O port buffer

On a standard Q-Monitor, the output buffer/transceiver at U31 is a 74LS244, which provides for 3 inputs and 4 outputs. If your application will require either of the alternate configurations (7 inputs or 7 outputs), remove the jumper from pins E23-E24 and insert it at E24-E25; also, ensure that the chip in U31 is correct for the selected configuration (see "Q-Monitor Reference Manual").

NOTE

Since the Q-Monitor uses some of its digital I/O lines for LTC, Failover Out and UPS operations, there are some restrictions on your use of the port:

- o if you are using the LTC or Failover Out features, you cannot configure the port as 7-inputs;
- o if you are are using UPS monitoring, you cannot configure it as 7-outputs

(see "Q-Monitor User's Guide" and "Q-Monitor Reference Manual" for details.)

1.4 Failover connector jumpers

With the standard jumper configuration (E26-E27, E30-E31), the Q-Monitor accepts inputs for A/D channels 2 and 4 on connector JA pins 7 and 4, respectively.

If you will be using "Failover", remove the jumper at E26-E27 and insert it at E27-E28; also remove the one at E30-E31 and insert it at E29-E30. In this case, JA-4 and JA-7 are used for the the Failover In/Out signals; A/D channels 2 and 4 can still be sampled using JB-13 and JB-14.

(See "Q-Monitor Reference Manual" and "Q-Monitor User's Guide" for further details of the Failover feature and for connector pin assignments).

2.0 When you plug it in ...

2.1 The Q-Monitor and the Q-Bus

The Q-Monitor can be installed in any standard Q-Bus slot in any 16-, 18- or 22-bit backplane, but it MUST NOT be installed in any slot with non-standard wiring, including any slots using the "C-D interconnect" scheme.

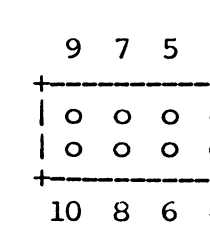
If you will be using interrupts for communicating with the Q-Monitor, or if you will be using Event interrupts (setting the interrupt bit in any Action code), you must confirm interrupt continuity between the processor and the Q-Monitor; i.e. all Q-Bus slots between the processor and the Q-Monitor must contain bus grant cards or other devices which pass BIAKI and BIAKO.

2.2 Connecting a terminal

To use QMsetup, you must initially attach a terminal to the Q-Monitor's serial I/O port (connector JC). This will let you verify the Q-Monitor's operation, and it will give you the chance to make any necessary firmware changes before running host-resident programs. Once the Q-Monitor is installed and configured to your satisfaction, you can disconnect the terminal if you do not intend to use it during normal operations.

The Q-Monitor's serial I/O connector mates with the same 10-pin connectors as the DLV11J:

Serial I/O Connector (JC)



pin		pin	
1	no connection	6	reserved
2	ground	7	ground
3	serial out	8	serial in
4	ground	9	no connection
5	no connection	10	no connection

Before connecting the terminal, set it to transmit and receive at 300 baud with a character format of 8 data bits, no parity bit. These are the Q-Monitor's defaults -- the characteristics in effect after a RECALL FACTORY command, or if the battery has been removed.

3.0 Before you boot your system ...

3.1 Powering the system up

When the Q-Monitor comes up, you may see

```
QMinit x
```

on the Q-Monitor's terminal if any non-volatile memory is uninitialized. You may disregard this message for now, but you should familiarize yourself with its implications later (see "Q-Monitor User's Guide" and "Q-Monitor Reference Manual").

Enter QMsetup and follow the instructions of the built in help messages: press the BREAK key, then type "?". Baud rates other than 300 can be selected with the SET BAUD command and you can use SET PARITY to select the alternate character format of 7 data bits plus an odd parity bit (no other formats are supported). To select 9600 baud for example,

```
QMsetup_ SET BAUD 2}
QMsetup_ SAVE
```

What is shown above seems to be fairly typical for one of our elderly VT100s, but it might look slightly different on your terminal. The "}" is a garbled character printed by the terminal when we typed RETURN at the end of the command line. That is what our VT100 (still running at 300 baud) thinks the string "QMsetup_" (now transmitted at 9600 baud by the Q-Monitor) looks like. At that point we changed the terminal's baud rate and then typed another RETURN, just to get the Q-Monitor to print a new prompt, and issued the SAVE command.

Remember that most parameter settings do not survive restarts unless you issue the SAVE command (CLOCK and ACTIONS are two notable exceptions).

At this point you may wish to use the SHOW CLOCK and SET CLOCK commands to set the Q-Monitor's calendar clock to local time, e.g.

```
QMsetup_ SHOW CLOCK                 show Mountain Standard Time
WED 28-JAN-87 13:12:19.3
QMsetup_ SET CLOCK 15:12:45         set Eastern Standard Time
                                    (SAVE is not needed here)
```

3.2 Interrupt priorities

Default Q-Monitor interrupt priorities are level 4 for serial I/O and level 5 for all Event interrupts; if your system will be using other interrupt priority levels, you should change these values now, e.g.

```
QMsetup_ SHOW PRIORITIES
4   5   5   5   5   5   5   5
QMsetup_ SET PRIORITIES 4 5 5 5 5 5 6 6   (level 6 for CT2, Failover In)
QMsetup_ SAVE
```

NOTE

- The Q-Monitor does not generate level 7 interrupts.
- Use of level 5 or level 6 interrupts for serial I/O may require modifications to your operating system.

3.3 Interrupt vectors

The Q-Monitor's default firmware configuration specifies four vectors: one for serial receive, one for serial transmit and two for Event interrupts. The defaults are (in octal):

Serial I/O		Events with data	Events without data
(330,332)	receiver	(274,276)	(270,272)
(334,336)	tranmitter		

If you plan to use interrupts for serial I/O with the Q-Monitor, you must ensure that there are no conflicts over the serial I/O vectors.

If you are using your Q-Monitor as the console terminal (not applicable to MicroVAX systems), you MUST change the serial I/O vectors to (60,62) and (64,66), and SAVE the new configuration before bootstrapping your operating system, e.g.:

```
QMsetup_ SHOW VECTORS
330 270 270 274 274 270 270 270
QMsetup_ SET VECTORS 60 270 270 274 274 270 270 270
QMsetup_ SAVE
```

Unlike bus address conflicts, there is no simple way to determine, a priori, whether or not your system contains a device which uses any of the Q-Monitor's default vectors. There are, however, three considerations which simplify the task:

1) the Q-Monitor's default serial I/O vectors (330,332) and (334,336) were chosen because they are in a "hole" not usually filled by DLV11Js;
2) the floating vectors used by DLV11-like devices are relatively few and generally are not used by most types of devices -- you should need to determine only the vectors used by other serial I/O devices (DLV/DZV/DZQ/DHV11s); and
3) the Q-Monitor's default Event vectors (270,272) and (274,276) are indicated in the Digital documentation as "user reserved" -- if your system contains no "unusual" hardware (other than the Q-Monitor itself), then there probably is no conflict over these vectors.

If you determine that your system does have another device using either of the serial I/O vectors, and if you will be using interrupt I/O with the Q-Monitor, then you should use QMsetup to change the Q-Monitor's serial I/O vectors prior to bootstrapping your operating system, e.g.:

```
QMsetup_ SHOW VECTORS
330 270 270 274 274 270 270 270
QMsetup_ SET VECTORS 370 270 270 274 274 270 270 270
QMsetup_ SAVE
```

NOTE

- The serial receive vector is the first one in the list.
- Serial I/O uses two vectors, but only the receive vector is specified; the transmit vector is always forced to the value of the receive vector plus 4.
- Because the Q-Monitor powers up with Event interrupts disabled, only the serial I/O vectors need be configured during installation; the vectors for the Event interrupts can be changed at any time prior to their actual use.

3.4 LTC

If you will not be using the Q-Monitor as the system LTC, you may reconfigure CT1 for use as an ordinary counter/timer at any subsequent time. However, we suggest that you issue the following commands at this time

```
QMsetup_ SET LTC 0
QMsetup_ SAVE
```

to free CT1 and digital I/O port line <4> for other uses.

If you will be using the Q-Monitor to provide the system LTC, you may need to change its frequency by specifying a different time constant at this point; you can do so with the command sequence

```
QMsetup_ SHOW LTC           default LTC
2500                         frequency is 50 Hz
QMsetup_ SET LTC x          set it as desired (x = 0 ... 65535)
QMsetup_ SAVE               make it permanent
```

NOTE

- o SET LTC actually sets the "half period" (in 4 usec units) of the LTC square wave; e.g. to generate the default 50 Hz LTC (square wave with a period of 20 msec), we set the time constant to 2500 for a "half period" of 10 msec (2500 x 4 usec).
- o Not all LTC rates can be generated precisely using the on-board 250 KHz clock; however, it is possible to drive CT1 with an external oscillator (see "Q-Monitor User's Guide").

3.5 Digital I/O port

If you are using a digital I/O port buffer chip other than the 74LS244 (i.e, if you are using other than the standard 3-in/4-out configuration), you must alter the Q-Monitor firmware accordingly, e.g.:

```
QMsetup_ SET PORT BUFFER 0          for 7 inputs, or
QMsetup_ SET PORT BUFFER 2          for 7 outputs
QMsetup_ SAVE
```

3.6 Failover monitoring

As shipped, the Q-Monitor's Failover feature is disabled. If you intend to use Failover, you may enable it at any time, e.g.

```
QMsetup_ SET FAILOVER CONFIGURATION L L     (set F'over In:  active Lo
QMsetup_ SAVE                                    F'over Out: active Lo)
```

4.0 After you've run it awhile ...

You may need to adjust the Q-Monitor's calendar clock oscillator or A/D converter voltage reference:

4.1 Adjusting the A/D converter voltage reference

The A/D converter uses an adjustable voltage reference. Verify that the reference voltage has not changed; if it has, readjust it as follows:

Attach a voltmeter to the side of R32 marked with the "+" in Appendix A. If, with the power on, the voltage is not +5.376, adjust trimpot R31: turn the screw clockwise to increase the voltage, counter-clockwise to decrease it.

4.2 Adjusting the calendar clock oscillator

An on-board 32768 Hz oscillator (Y3) drives the calendar clock chip. The calendar clock uses CMOS components which can be damaged by static electricity; also, capacitance can alter the frequency of the oscillator, particularly if the main power is off, so be careful not to touch U35 or nearby components.

The frequency of the oscillator is adjusted at the factory, but, if you have access to an accurate frequency counter, it is a good idea to check it (and adjust it if necessary) periodically.

NOTE

IMPORTANT

TP1 is the oscillator output (when the SET OSCILLATOR command is active); TP2 is the battery ground which, when main power is removed, is connected only to the oscillator, calendar clock chip, and CMOS RAM.

Let the frequency counter input float (i.e. do not connect the ground lead). If you were to power the system down with the counter grounded to TP2, system ground would be connected (through the counter) to battery ground and the lithium battery would try to power everything connected to the +5V line (the entire system).

To adjust the oscillator frequency:

1) Use QMsetup to gate the oscillator to TP1 ...
    QMsetup_ SET OSCILLATOR
2) Measure the frequency at TP1
3) Adjust the frequency, if necessary, by turning trimcap C34 with a plastic screwdriver. Adjust carefully, as an error of only 0.1 Hz causes a timekeeping error of 0.26 seconds per day
4) Return to normal operation by typing "R"
5) Use SET CLOCK to set the correct time

5.0 If you have trouble ...

and the trouble is ...	you should ...
(at any time)	
No response to BREAK	Verify terminal set to 300 (BAUD=7), or last value SAVED. Verify terminal set to 8-bits, no PARITY, or last value SAVED. Check cable pinout and integrity. Verify BREAK recognition is enabled using a host-resident program.
(when Q-Monitor is console terminal)	
No console ODT	Verify that Q-Monitor is not in QMdebug or QMsetup. Verify correct address PAL in U14.
Console ODT works, but operating system doesn't finish bootstrapping	Verify serial I/O VECTOR has been set to 60 and SAVED. Verify serial I/O PRIORITY is correct (SHOW PRIORITIES). Verify system LTC status (i.e. make sure that exactly one device is driving BEVNT)
(when Q-Monitor is not console terminal)	
Operating system shows "TTn" offline	Verify correct address PAL in U14.
Operating system shows "TTn" online, but can't talk to it	Verify that Q-Monitor is not in QMdebug or QMsetup. Verify correct serial I/O VECTOR, has been set and SAVED. Verify serial I/O PRIORITY is correct (SHOW PRIORITIES).

NOTE

Because of the Q-Monitor's myriad configuration possibilities, and because some Q-Monitor features cannot be made completely "docile", there remains a chance that you might inadvertently command the Q-Monitor into some mode that you will later come to regret. Should this occur, power the system down and momentarily slip a piece of stiff paper between the battery and its spring steel retainer. This causes a complete reset of the Q-Monitor, including the date and time and all behavioral parameters.

During initialization, the Q-Monitor displays a status code in RBUF (the receiver data buffer), which can be examined from the host system, either with console ODT (unless the Q-Monitor is the console terminal) or by a program. The appearance of these codes in RBUF generates neither an interrupt nor "Receiver Done" (RBUF<7>). Codes are displayed as follows (values are octal):

340	until Q-Monitor passes its stack write/readback test, then
300	until it passes its ROM checksum comparision, then
200 + QMinit code	until the first I/O operation overwrites RBUF, e.g. 217 corresponding to "QMinit F", or for normal starts, 200 when no QMinit message is printed

If the Q-Monitor fails either the stack write/readback test or the ROM checksum comparison, it freezes with the corresponding code displayed in RBUF, with its interrupts and Actions disabled, and with "Receiver Done" and "Transmitter Ready" negated (in other words, it "plays dead").

6.0 Before you call us ...

If, after following the procedures in this guide, you are unable to bring the Q-Monitor online and need to consult our personnel, you should:

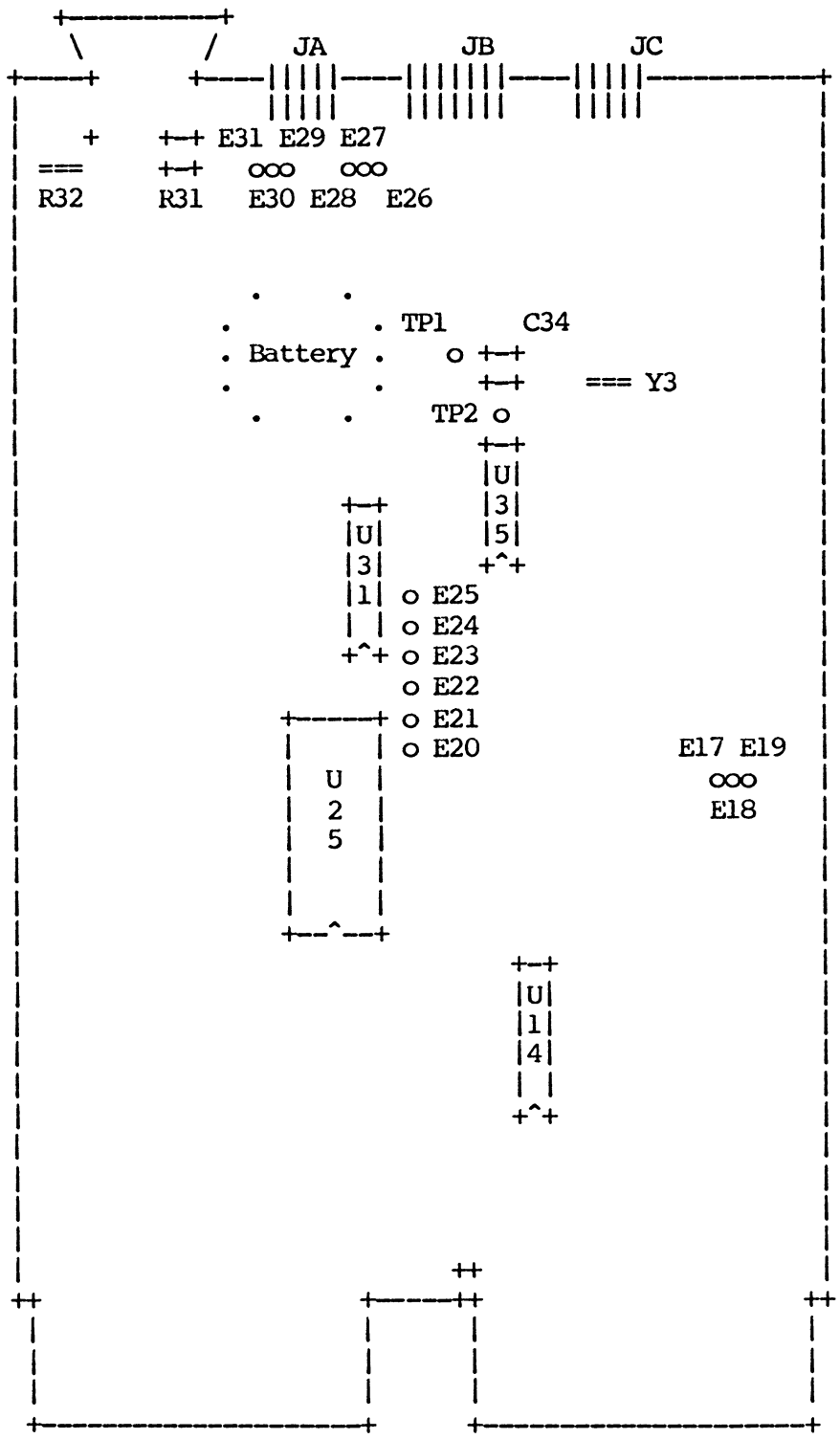
- examine RBUF and note its contents
- note the version number printed on the ROM (U25)
- note the values returned by QMsetup's SHOW ID and SHOW BUILD commands (if you can get into QMsetup)

We will also need to know the following about your system:

- type of enclosure (Digital BA23, Codar 600M, "homebuilt", etc.)
- type of processor (KA630, KDJ11-AC, etc.)
- operating system and version (RT11 V5.04, MicroVMS V4.4, etc.)
- configuration of your operating system's terminal driver
- serial number of the Q-Monitor
- configuration of the Q-Monitor, including jumpers and address PAL
- addresses and vectors of any DLV11-type devices
- vectors of any other devices using floating vector space (DZV11, DHV11, etc.)
- addresses and vectors of "unusual" hardware (A/Ds, custom devices, etc.)

Appendix A

Board Layout



BLANK

Separator

Separator

TABLE OF CONTENTS, PART I

1.0 Introduction 1-1
1.1 Getting Started
1.1.1 Terminology
1.1.2 Board Features

2.0 Setup Software 2-1
2.1 General
2.1.1 Set Commands
2.1.2 Show Commands
2.1.3 Special Commands

3.0 Hardware Organization 3-1
3.1 General
3.1.1 Q-Bus Interface
3.1.2 Z8 Microprocessor and Memory
3.1.3 Z8036 and Parallel I/O Port
3.1.4 Watchdog Timer
3.1.5 Calendar Clock
3.1.6 A/D Converter
3.1.7 Redundant System Port
3.1.8 Temperature Sensor
3.1.9 Battery Backup Circuit
3.1.10 Counter/Timers
3.1.11 RS232 Serial Interface

4.0 Installation 4-1
4.1 System Compatibility
4.1.1 External I/O
4.1.2 Board Adjustments
4.1.3 Powerup and Checkout

APPENDICES

A System Block Diagram
B Physical Layout Diagram
C Q-Bus Addresses and Bit Assignments
D I/O Port Configuration Guide
E External Connectors JA, JB, JC
F Counter/Timer I/O Guide and Timing Diagrams
G Jumper Pin Identification
H Technical Specifications
J Warranty
K Index, Part I

BLANK

INTRODUCTION

CHAPTER 1

1.1 GETTING STARTED

The Q-Monitor can be a complex board due to its many capabilities. The basic use of the board is not difficult, however, and it is not necessary that the user read this entire manual to begin using the Q-Monitor. Some sections of the manual are included for those users interested in details and advanced capabilities of the board, while other sections are essential for proper use of the board.

To get started, all users should read Chapters 1 and 5 of Part I and the first few pages of Part II.

If the user just desires to quickly verify that the board is working and/or use the Q-Monitor's serial port, then Chapter 4 (INSTALLATION) should be read.

1.1.1 TERMINOLOGY

In this manual when reference is made to digital I/O port, parallel I/O port, I/O port, or just port, the same port is being referred to. The word parallel is not to be taken to imply a port used for communicating with any sort of device such as a parallel interface line printer. The only other type of port on the board is a serial RS232 port (see Appendix A), and if that port is being referred to the word serial or descriptor RS232 will be included to avoid confusion.

The I/O port is physically 8 bits wide, but the user can never have access to more than seven of them. For this reason, when the discussion is centered on hardware issues, the port is referred to as a 7-bit port. When software issues are being discussed, the port is considered as an 8-bit or byte wide port even though only 7 bits are available. In this way we can speak of writing/fetching a byte of data to/from this port.

Reference will often be made in this manual to "host", as in host processor, host software, etc... Host in this case refers to the running Q-Bus processor. All references to resident or on-board mean hardware or software physically on the Q-Monitor board.

1.1.2 BOARD FEATURES

The Model 170 Q-Monitor provides extensive monitoring functions for Q-bus systems. The board monitors parameters critical to the operation of its host system, and (optionally) parameters of a remote or redundant system. The Q-Monitor takes appropriate action automatically when system parameters cross user-programmable thresholds or meet user-defined conditions. The action taken by the Q-Monitor is itself programmable.

In addition, the Q-Monitor provides a single serial port for use with a line printer or terminal.

The Q-Monitor may be partially programmed with the setup software (sec 2.0) which requires only that a terminal be connected to its RS232 port, or it may be fully programmed via its Q-Bus interface (Part II).

The basic features of the board are as listed below. This list is not exhaustive, and for a more detailed and complete discussion the hardware and software sections of the manual should be consulted.

Q-MONITOR BASIC FEATURES:

* checks system +5 volts, +12 volts, and temperature for excursions outside programmable limits

* a system watchdog with seperate, programmable delays for bootup and normal running

* calendar clock maintains date and time from .1 seconds through years with automatic leap day insertion.

* single RS232 serial port that operates with DEC software, including XON/XOFF protocol. Nine communication rates from 110 to 19,200 baud are program selectable.

* 7-bit parallel I/O port for driving or monitoring external lines. The configuration of the port lines (7 in, 7 out, or 3 in/4 out) is programmable.

* redundant system port for monitoring +5 volts, +12 volts temperature, and status of external systems that are also equipped with a Q-Monitor board.

* CMOS RAM, organized as 8K x 8, with battery backup. All critical sampled system data are stored here.

* two 16-bit counter/timers with external I/O for user access

* A/D converter accepts up to five channels of external inputs

* extensive ROM-based software provides above features and allows user to configure Q-Monitor to meet their particular requirements. Many system features are programmable either from terminal plugged into Model 170 serial port using interactive setup program, or via running Q-Bus software.

SETUP SOFTWARE

CHAPTER 2

2.1 GENERAL

The Q-Monitor's various functions are controlled by code residing in the 27128 EPROM. Most functions must be initiated by sending the desired command from the host Q-Bus processor, as described in Part II of the manual.

A limited number of commands, however, may be implemented by entering them on a terminal plugged into the Q-Monitor's RS232 serial port. The serial port is reached on connector JC, and the pin configuration for JC is detailed in APPENDIX E. This cable is not included with the Q-Monitor board.

Let's assume you have attached a compatible cable from your terminal to connector JC on the Q-Monitor. Your terminal must be initially set to 110 baud (this is to allow use by old, slow devices), but may be changed immediately after changing the Q-Monitor serial baud rate as detailed below. After the system is powered up, entry to the setup program is made by pressing the "break" key on your terminal.

At this point, you will get a QM_ prompt. Type a "S" and this will get you into the setup program with a QMsetup_ prompt. When you are done in setup, press the escape <ESC> key and you will return to the QM_ prompt. At this point, type P and you will return to the terminal emulation mode described further in Part II of the manual.

It is possible that upon powerup the message "NVRAM corrupt" will be displayed. This is not a problem, it just means that factory default parameters have been loaded from ROM because there is a problem with the ones saved in CMOS RAM. It is possible that during shipping there was a loss of data in the RAM. Press the break key and get the QM_ prompt as described above, enter setup by typing "S", and then save the current parameters (just loaded automatically from ROM) by entering the SAV CUR command described below under the heading SPECIAL COMMANDS. This command will store the current parameters and a checksum so that next time the system is powered up the RAM should not be "corrupt". Incidentally, the user may program a startup message, if desired. See the SET MESSAGE command in Part II of the manual.

The first command you will probably want to issue is SET BAUD (see

next page), so that you can operate at faster than 110 baud.

COMMANDS

From the setup software, the following commands may be executed and the syntax for each is listed below. If you make a mistake typing, a backspace or delete will erase one character or ^U (control U) will erase the entire line. All command lines must be terminated with a carriage return. Further information on most of the commands listed below may be found in Part II of the manual.

****** IMPORTANT NOTE ******

In a few cases, the format for the commands below is slightly different from that listed in Part II of the manual. When using the on-board setup, the format must be as listed below.

NOTE: All of the commands listed below consist of two words, a verb and a noun. All verbs may be abbreviated to to the first two letters, all nouns to the first three.

2.1.1 SET COMMANDS

1) SET CLOCK

To set the correct date and time for the Q-Monitor's calendar clock (for example, Monday, January 1, 1985 at 8 o'clock in the morning) type:

type SET CLOCK MON 01-JAN-85 08:00:00 , then carriage return.

notes:

1) leading zeroes must be typed, as shown above
2) time is military style, hours 0-23.
3) days and months are abbreviated as follows:

MON, TUE, WED, THU, FRI, SAT, SUN
JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG
SEP, OCT, NOV

2) SET PARITY

Parity may either be odd or disabled. Type:
SET PAR O to set odd parity, SET PAR D to disable parity

3) SET BAUD (RATE)

Refer to Part II of the manual to get the correct single digit code for the nine settable baud rates. To set the board to 9600 baud, for example, type SET BAU 2

4) SET VECTORS (INTERRUPT)

Refer to Part II of manual for information on the eight vectors. The vectors must be entered as two hex digits, and all eight must be entered even if only some are changing or being used. For example, to set the serial line unit receive interrupt vector to 330 (octal) and not use any of the other eight, type:

```
SET VEC D8 B8 B8 B8 B8 B8 B8 B8
```

The serial line unit transmit interrupt vector is automatically taken to be the receive interrupt vector + 4. Note that D8 (hex) is 330 (octal) and the hex B8s are merely dummy values that must be entered as legal values, even though those vectors are not used in this example.

5) SET PRIORITY

Refer to Part II of manual for a description of Q-Bus interrupt priority for the vectors described above. For example, suppose you wish to set the serial line unit priority to 4 and not use any other vectors. Again, you must enter a legal value for the others anyway. In this case you could enter a value of 4 for all eight priorities by typing:

```
SET PRI 4 4 4 4 4 4 4 4
```

6) SET BUFFER (PORT)

NOTE: This is same as host SET PORT BUFFER command

There are three choices here. SET BUF 0 configures the parallel I/O port to be 7 input lines, SET BUF 1 configures 3 inputs/4 outputs (the default setting), and SET BUF 2 configures 7 outputs. Note that the standard Q-Monitor as shipped is only compatible with SET BUF 1. If another configuration is desired, the socketed IC U31 will have to be changed, as detailed in Appendix D.

7) SET TERMINATOR

Refer to Part II of manual for a description of the reply terminator. For example, to set the terminator to an "}", type:

```
SET TER 7D
```

where 7D is the ascii hex code for the right curly bracket.

8) SET OSC

This command allows user to make adjustments to the calendar clock oscillator freqency. See page 5-2 for details. The command is entered by typing SET OSC and can only be exited by typing "R" for resume. There is no SHOW OSC command.

2.1.2 SHOW COMMANDS

For all of the above "set" commands, except "set osc", there is a corresponding "show" command, which will show the "in-use" value (see note in 2.1.3 below) of the appropriate parameter. These are defined in Part II of manual. To show the current date/time, type:

SHOW CLOCK

As another example, to show the current interrupt vectors, type:

SHOW VEC

In a similar fashion, all the other show commands may be implemented.

2.1.3 SPECIAL COMMANDS

1) REC FAC

This command (recall factory) causes the factory default parameters to be loaded from ROM into the operational page of CMOS RAM. The operational page is where the parameters currently being used are stored.

NOTE: Executing this command, however, does not cause immediate usage of all items in the newly loaded operational page. Some parameter values are checked just once upon powerup, while others are checked as needed. The only way to make those in the latter case take effect immediately is to issue the proper individual command from the host or setup program, as appropriate.

For example, the baud rate may be changed immediately by issuing the SET BAUD command as described above, or by host command, but executing a setup REC FAC command (where the factory value is different) will not cause immediate change. This is so even though a setup SHOW BAUD command would show the factory value. This restriction will be removed in the future. At the present time, if in doubt, issue the individual command.

All the individual "set commands" listed above make modifications to the operational page.

2) REC SAV

This command (recall saved) causes "saved" parameters in CMOS RAM to replace the set of parameters currently being used by the Q-Monitor (those in the operational page). See the comment above under REC FAC for parameter usage.

3) SAV CUR

This command (save current) causes the current parameters in use to be saved in CMOS RAM for later use. Thus if changes are made to some parameters, and then this command is executed, the next time the board is powered up the saved values will be used.

BLANK

HARDWARE ORGANIZATION

CHAPTER 3

3.1 GENERAL

The basic hardware structure of the Q-Monitor is as shown in the block diagram in Appendix A.

The Q-Monitor occupies a single dual-wide slot in the Q-Bus and is compatible with all 16, 18, or 22-bit Q-Bus systems.

The Q-Monitor's basic operation is controlled by a Z8 microprocessor executing a maximum of 16 K x 8 of ROM-based code. In addition, there is 8k x 8 of CMOS RAM for data storage. A Zilog Z8036 counter/timer and parallel I/O unit helps provide the interface to the Q-Bus and to the external world, as well as providing internal timing functions.

Timekeeping functions are provided by a National Semiconductor MM58274 and a National Semiconductor ADC0808 A/D converter digitizes analog data as required. A LM35CZ temperature sensor from National provides temperature data.

In addition to its Q-Bus interface, the Q-Monitor has three connectors (JA, JB, and JC) on its top edge for I/O.

For additional details, please refer to the system block diagram and physical layout diagram in Appendices A and B.

3.1.1 Q-BUS INTERFACE

a) addresses

The Q-Monitor appears in the I/O page and to the operating system looks similar to a single channel DLV11J serial interface. The board decodes four consecutive word addresses for register usage. The register standard addresses and bit assignments are shown in Appendix C.

These addresses are the standard factory assignments, and may be changed at time of order. Changes later may be made by ordering new address decode PALs.

The main difference from the DLV11J format is that this board does not support the reader enable bit in the RCSR, the error bits in the RBUF, or the transmit break bit in the XCSR. In addition, the Q-Bus watchdog timer function utilizes a read of the XBUF (normally an undefined operation) to reset itself and prevent a timeout response.

The Q-Monitor also uses bit 0 of the XCSR to provide resynchronization when necessary, between the Z8 and Q-Bus processor when commands, as opposed to serial interface data, are sent.

For more information on the register functions, consult the DEC literature.

b) interrupts

Since the Q-Monitor uses Q-Bus interrupts for purposes in addition to those required for the serial line interface, the full 4-level Q-Bus interrupt protocol is implemented. The interrupt vectors are a user-settable feature, so if the standard factory vectors produce a conflict in the user's system, it is easily resolved. The vectors are changed by using the setup software or by host command. Interrupt priorities 4, 5, or 6 may be selected in the same manner (priority 7 is not allowed by current version of Q-Monitor). The Q-Monitor puts an eight bit vector out to the Q-Bus, thus the maximum vector is 374 octal (must be divisible by 4).

c) data path

The data path between the Q-Monitor and the Q-Bus is eight bits wide. Data is accepted from the Q-bus by the Z8 at its Port 2, and all data sent to the Q-Bus (including interrupt vectors) first passes through Port A of the Z8036 CIO chip.

d) Q-Bus control lines

The Q-Monitor is capable of halting or rebooting the Q-bus processor by controlling the BHALT and BDCOK lines. These actions may occur in response to some event such as an overtemperature condition. For further details, see Part II of manual.

A reboot of the Q-Bus processor will likewise cause the Z8 to reboot unless the Q-Bus reboot was initiated by the Q-Monitor.

The Q-Monitor does not perform any Q-Bus DMA operations and the corresponding control lines are passed through.

3.1.2 Z8 MICROPROCESSOR AND MEMORY

The Q-Monitor uses a Z8681 microprocessor to control the many functions that the board is capable of performing. The Z8681 is a member of the Z8 family of processors and will be referred to from now on as the Z8. The Z8 is an 8-bit processor with a multiplexed address/data bus and a flexible arrangement of I/O lines.

The Z8 is clocked by an internal oscillator that is controlled by an external 7.3728 MHz crystal. The Z8 also has an internal serial line unit that is clocked by one of the Z8's internal counter/timers, and this crystal frequency allows easy derivation of nine standard baud rates from 110 to 19,200 baud.

Program memory on the Q-Monitor consists of a 16k x 8 27128 EPROM. Data memory is a 8k x 8 CMOS RAM whose contents are protected by a battery backup ciruit. The RAM will retain its contents for approximately 2 years with no system power.

During normal operation the Z8 maintains a data log of all important parameters in the CMOS RAM. The data log is continuously updated and in the event of a power failure (itself selectable as a logged event) all data is therefore safe. Data on the backup battery itself is also logged. See Part II of manual for more details on operation of the data log.

The Q-monitor has a watchdog timer circuit that prevents a spurious event from permanently disrupting operation of the Z8. This is in additon to the primary two-stage watchdog circuit that monitors the Q-bus processor. The Z8's own watchdog will attempt to reboot the Z8 if program execution is significantly disrupted (a designated address is not read every .5 seconds) and if this occurs the event is logged.

The Z8 reboot will not cause the Q-Bus processor to reboot. All future references to watchdog timer will mean the Q-Bus watchdog timer, unless otherwise stated.

3.1.3 Z8036 (CIO) and PARALLEL I/O PORT

a) Z8036

The Q-Monitor uses a Z8036 (CIO) to expand the I/O capabilities of the board. The CIO has four I/O ports, one of which interfaces it to the Z8 from which it receives all commands. Another port interfaces to the Q-Bus, and handles all outgoing 8-bit data including interrupt vectors. A third port provides the Q-Monitor with an 8-bit digital I/O connection to the outside world. The fourth port is used primarily for on-board timing purposes.

The CIO has an external 500 kHz oscillator that runs its three inter-

nal counter/timers (after an internal divide-by-two). One of these provides the Q-Monitor with a 10 Hz clock that is used internally for various things, including sampling the A/D channels and updating the data stored in non-volatile RAM.

Depending on the I/O port configuration, none, one, or both of the remaining two counter/timer outputs may be brought off the board through connector JB. One of the counter/timers may be set to take its gate,trigger, and counter inputs from external sources through the digital port (see below). In addition, there is the option of using one of these counter/timer outputs to provide the Q-Bus line time clock.

b) I/O Port

Port B of the CIO provides a 7-bit digital I/O connection to external devices through connector JB. The port is programmable by the user to be seven bits of output, seven bits of input, or three bits in and four bits out. Port B lines pass through a buffer/transceiver chip before they come out on connector JB. The chip location is socketed so that the user may change the configuration and/or type of output (open collector /tri-state, true/inverting) by just changing the chip and using the appropriate commands to record the change. If no signal inversions are involved the port may be configured with just the setup software.

Instead, if the user knows ahead of time that a chip other than the standand will be required, he may in some cases order the Q-Monitor with the optional chip.

Appendix B gives details on which output chips may be used for the three possible port configurations, chip characteristics, and the standard chip shipped with the board.

The user may program the Q-Monitor via host command to take specific action on a particular bit pattern (inputs and outputs, levels or transitions). If action is to be taken on input bits, those bits must be driven at the appropriate times by the user's hardware and not allowed to float. If the Q-Monitor is programmed to powerup with input bits on the I/O port, then those bits should be driven from that time on. If a change is made is a running Q-Monitor to include input bits, then the bits should be driven before the change is made.

Port C of the CIO provides an output line which passes through an open-collector buffer and also comes out on connector JB. This may be used as a (small) relay driver, for example, to drive an alarm when certain conditions such as overtemperature occur, without regard to the I/O port configuration. Host command is used to select conditions to drive this line active.

3.1.4 WATCHDOG TIMER

The shortest interval possible is .1 second, and the longest is 1:49:13.6. The watchdog timer is implemented in software; for more details see Part II of manual.

3.1.5 CALENDAR CLOCK

The Q-Monitor utilizes a National Semiconductor CMOS MM58274 calendar clock chip to provide timekeeping functions. The chip is capable of keeping the time from .1 seconds thru years, with automatic leap day insertion. The calendar clock circuit is protected against main system power failure or interruption by a battery backup circuit. The calendar clock will maintain the correct time for a maximum of two years with no main system power.

The calendar clock may provide the operating system with the correct time upon bootup (see Part II of manual) and in addition enables the Q-Monitor to store the time when certain events occur. For example, the Q-Monitor may be programmed to log the time of activation of a UPS system. See Part II of manual for a complete programming details.

The calendar clock chip derives its time base from an external 32.768 kHz oscillator circuit. An external oscillator circuit is used rather than the on-chip one in order to provide better frequency stability when in the the battery backup mode.

The date and time may be read or set at any time when the board is operating on main power, using either the QMsetup program or host software.

3.1.6 A/D CONVERTER

The Q-Monitor has an 8-channel (multiplexed) 8-bit A/D converter that is used to monitor the system +5 volts, +12 volts, and temperature sensor output. The remaining five channels may be employed by the user as detailed below.

The A/D is a National Semiconductor ADC0808 that operates from a +5.38 supply voltage derived from a National Semiconductor LM385BXZ precision voltage reference. The positive reference voltage for the A/D also comes from the precision reference.

The five A/D channels left for user inputs may be used as follows:

1) If the board is configured to use connector JA for redundant system operation, then three of the five remaining channels are automatically taken to monitor external +5 volts, +12 volts and temperature sensor outputs (from another system).

The remaining two channels (channels 2 and 4) may accept external inputs on connector JB (see Appendix E)

2) If the board is not configured for redundant system monitoring, then all five remaining channels may accept external inputs on connector JA, or if desired, take three external inputs from JA and two from JB (see Appendix E).

The A/D converter accurately converts voltages between 0 and +5.35 volts. This gives the converter a resolution of 21 mV per least significant bit. The supply voltage for the A/D is derived from a precision voltage reference running off the system +12 volts. This allows the Q-Monitor to track the system +5 volts past the DEC high specification of +5.250 volts.

The user must make sure that all external A/D inputs are within a range of -.3V to +5.65 V (accurate conversion range is less) or damage to the converter will occur. For this reason the Q-Monitor divides its own +12 supply. Since the A/D is a high impedance device (see Specifications) and has only 21 mV of resolution per LSB, a simple resitive voltage divider may be used to attenuate voltages larger than +5.35 volts if each resistor is no bigger than a few thousand ohms.

The channels of the A/D are multiplexed, and on a continuous basis each active channel is sampled once about every 80 msec.

The design of the A/D converter is such that all analog inputs must be essentially DC voltages (< 10 Hz) to avoid sampling errors greater than one LSB.

Following is a list of A/D channel assignments and where these channels appear on I/O connectors JA and JB.
(RS stands for redundant system)

channel (0-7)	A/D pin	channel use	connector pin assign
0	U42-26	RS +12/Analog in 0	JA-10
1	U42-27	RS + 5/Analog in 1	JA-8
2	U42-28	Analog in 2	JA-7 or JB-13 (no RS) JB-13 (RS)
3	U42-1	Analog in 3	JA-6
4	U42-2	Analog in 4	JA-4 or JB-14 (no RS) JB-14 (RS)
5	U42-3	+12 volts	JA-1
6	U42-4	+ 5 volts	JA-3
7	U42-5	temperature sensor	JA-5

3.1.7 REDUNDANT SYSTEM PORT

Two Q-Bus systems, each equipped with a Q-Monitor, can form a redundant system by connecting the Q-Monitors together.

The Q-Monitor always monitors its own system's +5 volts, +12 volts and temperature. When part of a redundant system, it also checks the same quantities from the other half of the system. In addition, each Q-Monitor has a control line (failover out) that it drives and provides to the other Q-Monitor which gives its status (running OK or not). This status line can be programmed to reflect any of a number of different conditions affecting the health of the host Q-Monitor.

Q-Monitors are connected together with a 10-conductor cable that plugs into header JA. The cable is run from one board to the other with a 180 degree twist in the cable. Therefore, JA-1 from one Q-Monitor connects to JA-10 on the other, JA-2 connects to JA-9, and so forth.

The +5 volts output from one Q-Monitor to the other is just the system +5 volts in series with picofuse F2. The " +12 volts " output is really +4 volts (to stay in range of A/D) after attenuation by a voltage divider, and in series with picofuse F1. The temperature output is driven by a LM385 opamp with a gain of 7. The user should verify that the cable connecting the two Q-Monitors does not cause problems such as oscillation or excessive voltage drop. See Appendix E for a diagram of the connector JA used with the redundant system.

NOTE:

when the Q-Monitor is configured for redundant system operation, the digital I/O port must be configured for either 7 bits output or 3 bits in / 4 bits out, because the failover out line is driven from Port B of the CIO chip.

3.1.8 TEMPERATURE SENSOR

The Q-Monitor uses a National Semiconductor LM 35CZ linear temperature sensor to track the system temperature. This device outputs a voltage proportional to the temperature in degrees Celsius. It is followed by a buffer amplifier with a gain of seven. The Q-Monitor is designed to check on the system temperature over a range of 0-70 degrees C. Over this range the output of the buffer amplifier, U41 pin 1, goes from approximately 0 to +5.03 volts.

The buffered temperature output goes directly to the A/D channel 7, and after passing through a 3 kHz low pass filter, to connector JA-5 for possible redundant system monitoring. If the errors due to the temperature sensor, the op amp, and the A/D are taken into account, then the temperature as read by the Z8 is accurate to about plus/minus three degrees C.

The temperature sensor is mounted directly over a small hole in the PC board, so that good airflow around the sensor is maintained.

3.1.9 BATTERY BACKUP CIRCUIT

The Q-Monitor employs a battery backup circuit to ensure that if the main system power fails or is interrupted, the calendar clock continues to keep the correct time and the data in the CMOS RAM is retained.

When the +5 volt supply falls below about +4.2 volts, the battery backup circuit disconnects the calendar clock and CMOS RAM ground leads from the PC board ground plane and applies the battery +3 volts (approx.) to the +5 volt supply leads. The battery is a 3 volt, type 2430 lithium button cell.

The calendar clock and CMOS RAM are then the only devices connected to battery ground. The calendar clock continues to operate and the RAM retains all data, both devices together drawing only about 10 uA from the battery. This current is low enough to allow both devices to continue in this way for about two years.

While backup battery power is applied, no writes can take place to either the clock or RAM because their write circuitry is gated off. This ensures that neither device will have its data corrupted by spurious signals as the power is failing.

When main power is restored, the backup battery automatically disconnects itself from all circuitry.

3.1.10 COUNTER/TIMERS

The CIO chip on the Q-Monitor has three internal 16-bit counter/timers. One of the counter/timers is used on a permanent basis by the board software to provide a 10 Hz clock for updating the data log and timing other functions.

The other two counter/timers (CT1 and CT2) may be employed by the user in a variety of ways. In timer mode, the counters run off a 250 khz clock derived from an on-board external 500 khz oscillator. In counter mode, CT1 and CT2 accept external user inputs including external clock. The user may have more complete access to the signal lines of CT1 than to those of CT2; see the discussion below.

Many features of the counter/timers are programmable, such as output frequency, output waveshape, single or continuous run, internal or external inputs, and others. CT1 and/or CT2 reaching terminal count can be programmed to be an "Event" in which case certain "Actions" take place, such as interrupting the host processor. All counter/timer programming is done by host command; see Part II of manual for details. For example, the counter/timer(s) are loaded with the value to be downcounted from using the SET TIMER CONSTANT command(s). All commands relevant to counter/timer operation are to be found in sections 14.75 through 14.84 in Part II of the manual, and also in Chapter 5, Part II.

External access to the inputs/outputs of CT1 and CT2 is affected by the fact that the port through which access is achieved may be used for other functions as well. For the following discussion refer to Appendix F. The results may be summarized as follows:

1) The parallel port is configured as 3 bits in / 4 bits out. In this case, the user has complete access to all CT1 inputs and outputs. The user has access to only the CT2 output signal. (See APPENDIX F-1).

2) The parallel port is configured as 7 bits in. In this case, the user has access to the gate, trigger, and counter in lines of CT1 and the trigger and counter in lines of CT2. (See APPENDIX F-2).

3) The parallel port is configured as 7 bits out. In this case, the user has access to the counter output lines of both CT1 and CT2. (See APPENDIX F-3).

****** NOTE ******

It is important to realize that the use of the counter/timers must be consistent with the other uses of the parallel port. For example, if the user has two Q-Monitors connected together as part of a redundant system, the signal line I/O 1 which appears on connector JB-3 is used as the failover output line. This requires that the port be configured as either 3 in / 4 out, or 7 out. It is clear that in case the port could not be configured for 7 bits in, as in 2) above, and the use of CT2 would be affected.

It is also important to make sure if the port configuration is changed, the port buffer chip is correct. It is also necessary, if making a configuration change, to program it either by host command or setup command (see Appendix D).

Some Possible Modes of Operation

1) CT1 generates LTC

The port configuration is the standard 3 inputs/4 outputs. CT1 may be used in timer mode to provide the Q-Bus with a line time clock (LTC) that may run at the usual 60 Hz or else at a user-programmable frequency. This frequency may range (in principle) from .0019 hz to 125 khz if using the counter/timer's on-board 500 khz oscillator. The Q-Monitor at present does not have a provision for a LTC to operate automatically, the board must be programmed to provide it. See the discussion below and then Chapter 5, Part II of this manual.

The usual 60 hz LTC is not a good choice if the Q-Monitor is to provide it, due to the error involved in dividing down the on-board 500 khz oscillator that drives the counter/timers. This error amounts to about 14 seconds/day or about 7 minutes per month. If the user can live with this large error then the 60 hz LTC can be generated by replacing the SET TIMER 1 CONSTANT 2500 instruction in Chapter 5, Part II of this manual with the same instruction but with 2500 replaced by 2083.

The Q-Monitor can provide a much more accurate LTC at a frequency of 50 hz. The error generated at this frequency is about .7 sec/day or about 20 seconds/month. Chapter 5, Part II of this manual details the necessary commands.

The user could put CT1 in counter mode and furnish an external clock to drive CT1 and the LTC, but there is little practical reason for doing so. See Installation, Chapter 4, regarding LTC usage. The frequency range quoted applies for a square wave output only, please see waveshape/frequency discussion below.

Since the port configuration is 3 inputs/4 outputs, CT2 is only usable in timer mode, with the on-board 500 khz oscillator. CT2 may, if desired, be configured for external output.

2) CT1 AND CT2 INDEPENDENT OPERATION

This mode is the same as 1) above, but CT1 is not providing a LTC. The port configuration is the standard 3 inputs/4 outputs. No LTC is being furnished by CT1. CT1 may be in either counter or timer mode, CT2 is in timer mode only. CT1 and CT2 may both output signals on JB-5 and JB-1, respectively. Each counter/timer is independently programmable.

3) CT1 AND CT2 CASCADED

CT1 may be cascaded with CT2 to provide extremely slow output frequencys. In this mode the output of CT1 drives the input of CT2. The necessary connections are made internally on the CIO chip, all that is required from the user is the proper programming of CT1 and CT2. The minimum output frequency is 2.91 x 10 -5 Hz using the on-board oscillator, and could be even slower if CT1 was driven externally. The output signal is CT2 out, and appears on JB-1.

When CT1 and CT2 are cascaded, it is still possible to have the CT1 counter output appear as an external output on JB-5. This is done by ensuring that the CT1 external output enable bit is set. The port configuration may be either the standard 3 inputs/4 outputs, or optional 7 outputs, depending on how CT1 is controlled.

External Inputs

NOTE: refer to Appendix F for discussion below

a) counter input - the clocking signal for the counter(s). It must not exceed 2 MHz.

b) trigger input - on the rising edge of this signal the counter loads the user selected count. On the next rising edge of the counter input the counter begins its down count, if timing constraints have been observed.

c) gate input - this input enables the counter. As long as it is high, the counter counts if all other conditions are met.

If the gate goes low, the counter will stop operation if it was counting, and resume operation from where it was when the gate goes high again. The gate does not affect the trigger.

Continuous/Single Cycle

The user may select either continuous or single cycle mode for either counter/timer.

Output Waveshape and Frequency

In addition, the user may select the output waveshape for external counter/timer output where appropriate. The outputs may be one of the following:

a) square wave
b) pulse
c) one-shot

see Appendix F, page F-3 for diagrams of all three waveshapes

Note:
The output frequency obtained with the pulse output is simply the 250 kHz internal clock frequency or the user supplied count input frequency divided by the 16-bit binary number loaded into the counter (non-cascaded operation). Loading a zero value for this 16-bit quantity results in a count of 65,536 being loaded, the maximum available. With the square wave output, the frequency is half that of the pulse output, because of the way the CIO chip produces this output. Thus for single counter operation the output frequency range is:

i) square wave - .0019 Hz to 125 kHz
ii) pulse - .0038 Hz to 250 kHz

All three external inputs must be TTL-compatible. Detailed timing diagrams are provided in Appendix F which cover all timing considerations.

3.1.11 RS232 SERIAL INTERFACE

The Q-Monitor provides a single channel RS232 serial interface between the user's terminal and the Q-Bus. This interface is capable of running at nine standard baud rates from 110 to 19,200 baud. The default baud rate is 110, other baud rates may be selected either temporarily or permanently using the setup software.

The RS232 interface is implemented using "data leads" only, that is, transmit data, receive data, and ground. The terminal is connected to the interface on connector JC (see Appendix E). The recommended cable receptacle is the AMP PN 87133-5 or equivalent. If this part is used, three AMP PN 87165-1 locking terminal clips will be required for the three signal leads.

The UART for the interface operates with one or more stop bits for re-

ceive data and supplies two or more stop bits for transmit data. The UART supports either odd parity detection/generation or no parity.

The interface supports break detection as a means of entering several Q-Monitor operational modes. Among these are front panel, debug, set-up, and terminal. For more details, see section 5.2.

INSTALLATION

CHAPTER 4

4.1 SYSTEM COMPATIBILITY

This section of the manual discusses things that the user must do to make sure that the Q-Monitor will function correctly when installed in his Q-Bus system. The Q-Monitor is compatible with all 16, 18, or 22 bit Q-Bus systems. On MicroVax systems (and other systems using a "C-D backplane") care must be taken not to plug the Q-Monitor into any of the C-D slots.

***** NOTE ON BOARD HANDLING *****

The Q-Monitor is shipped with a non-conductive plastic sheet glued to the bottom of the board. The purpose of this sheet is to prevent the Lithium battery from shorting out if the board is placed on a conductive surface. For this reason the sheet also protects the battery backup operation of the calendar clock circuit and the CMOS RAM. If the plastic sheet is removed, it must be re-glued with RTV non-corrosive adhesive. Corrosive RTV can usually be identified by a strong vinegar-like acetic acid smell. See also the comments below under CALENDAR CLOCK OSCILLATOR regarding board handling.

a) BAUD RATE

If the user intends to plug a terminal into the Q-Monitor's serial port, he must make sure his terminal is set for 110 baud when the board is first powered up (Q-Monitor default setting). Other baud rates up to 19,200 baud may then be selected for temporary or permanent use using the setup software or sending the appropriate command from the host processor. The Q-Monitor powers up with parity disabled.

b) INTERRUPTS

Proper specification of Q-Bus interrupt vectors is essential if host software will be using interrupt I/O with the Q-Monitor; the user must make sure that there is continuity on the interrupt lines between the processor and Q-Monitor.

c) INTERRUPT VECTORS

There are as many as nine different interrupt vectors that the Q-Monitor may use. The board as shipped from the factory is programmed to use only two, for the serial port, and the default values are:

1) receive interrupt 330 (octal)
2) transmit interrupt 334 (octal)

It is possible to change the above vectors if they are already being used by means of the setup software (sec 2.0) or by issuing a command from the host processor (see Part II of manual).

The user should make sure that no other devices use these vectors. If the Q-Monitor is programmed to use any additional vectors then these too, should be checked.

d) ADDRESSES

The Q-Bus addresses that the standard Q-Monitor uses are listed in Appendix C. The user should verify that no other device on the bus uses any of these addresses. These addresses are not changeable under program control, if another address such as the console terminal address is desired, special address decode PALs must be ordered from CODAR.

e) LINE TIME CLOCK

If the Q-Monitor is to provide the system LTC, there are two jumpers that must be set (see Appendix G) and the appropriate command must be sent by the host processor to the Q-Monitor. The user should verify that no other devices output a LTC to the Q-Bus.

f) JUMPERS

The Q-Monitor has six features that the user may configure with jumpers. Several other jumpers are factory installed and are not to be altered. See Appendix G for jumper locations and functions.

4.1.1 EXTERNAL INPUT/OUTPUT

a) CABLES/CONNECTORS

The Q-Monitor interfaces to the outside world with three connectors:

1) A serial device (terminal or printer) if used with the Q-Monitor plugs into the 10-pin connector JC. See section 3.1.11 for recommended cable fittings.

2) The 7-bit I/O port is accessed through the 14-pin connector JB. A/D channels 2 and 4 may (optionally) accept external inputs on JB also. All standard ribbon cable types will fit.

3) The redundant system port is reached on the 14-pin connector JA. If JA is not used to monitor a redundant system, it may be used to

provide access to five channels of the A/D converter. All standard ribbon cable types will fit.

NOTE:
See Appendix E for connector pinouts and signal descriptions

b) OUTPUT BUFFER

The output buffer/transceiver chip U31 is a 74LS244 for a standard factory configured board. If the user reconfigures the parallel I/O port (see Appendix D) he must ensure that the chip in U31 is correct for the selected configuration.

4.1.2 BOARD ADJUSTMENTS

a) A/D CONVERTER ADJUST

The A/D converter operates off an adjustable voltage reference that is set at the factory to +5.376 volts. The user should verify that the reference voltage has not changed, and if it has, readjust it. This is done as follows:

1) Attach a voltmeter to the side of R32 marked with the "+" on the physical layout diagram (see Appendix B). If the voltage read at this point with the power turned on is not correct, then proceed to (2).

2) Adjust trimpot R31 until the voltage is correct. Turning the screw clockwise should increase the voltage, turning it counter-clockwise should decrease it.

b) CALENDAR CLOCK OSCILLATOR

An external 32.768 KHz oscillator is used to drive the calendar clock chip. The oscillator uses CMOS components, so if your climate is conducive to the buildup of static electricity, be careful not to touch U35 (see Appendix B) or nearby components with your hands. Static discharge may cause permanent component damage, and even if it doesn't the frequency of the oscillator may be altered by the presence of your hand, particularly if the main power is off.

The frequency of the oscillator is adjusted at the factory, but if you have access to an accurate frequency counter it is a good idea to check and adjust the frequency, if necessary, upon receipt of the board.

NOTE

It is best to do this with main power applied to the board. If it is necessary to do this with the main power off and the battery backup circuit activated, make sure that the frequency counter inputs are floating (not referenced to ground).

Otherwise, system ground is connected to battery ground and the lithium battery will try to power everything connected to the +5 V line (most everything). TP1 is the oscillator output, and TP2 is the battery ground, which is a local ground connected only to the oscillator, calendar clock chip, and CMOS RAM when main power is removed. See also section 3.1.9.

To check/adjust oscillator frequency:

1) Enter QMsetup (see section 2.0) and enter the SET OSC command. The oscillator output (buffered) will then appear on TP1. Measure the frequency between TP1 and TP2.

2) Adjust the frequency, if necessary, by turning trimcap C34 with a non-metallic screwdriver. Adjust carefully, as an error of only 0.1 Hz causes a timekeeping error of .26 seconds per day.

3) Terminate the SET OSC command by typing "R". Using the SET CLOCK command , set the correct time. Then use the SHOW CLOCK command several times to verify that the clock has restarted and has the correct time. If desired, leave QMsetup by typing <ESC>.

4.1.3 POWERUP AND CHECKOUT

The easiest way to verify quickly that the Q-Monitor is working correctly is to plug a terminal into the serial port and perform a few operations. Details on the serial port may be found in sec 3.11 and details on connector JC pinout may be found in Appendix E.

Before plugging a terminal in and powering up, the user should have first read first few pages of Part II and Setup Software, Chapter 2 of Part I. Assuming this has been done, power the system up and after your system has booted up, you should get your system prompt by just typing a carriage return. At this time you should be able to login to your system and proceed just as with a normal serial port such as a DEC DLV11.

Next, press the "break" key on your terminal, if it has one, or type the equivalent combination of keys if it doesn't. The Q-Monitor will issue a QM_ prompt and wait for operator input. At this point you have four choices of a single key entry:

1) D is debug mode
2) F is front panel mode
3) P is proceed command
4) S is setup mode

Only setup mode will be discussed here, for a description of the other three modes, see Part II of manual.

Type S and get into setup mode. The prompt QMsetup_ will appear. Try the SET CLOCK and SHOW CLOCK commands as detailed in sec 2.0. You

should be able to set and show the date kept by the board's calendar clock. Although the date is set at the factory, it may not be correct when you receive it due to shipping and handling effects.

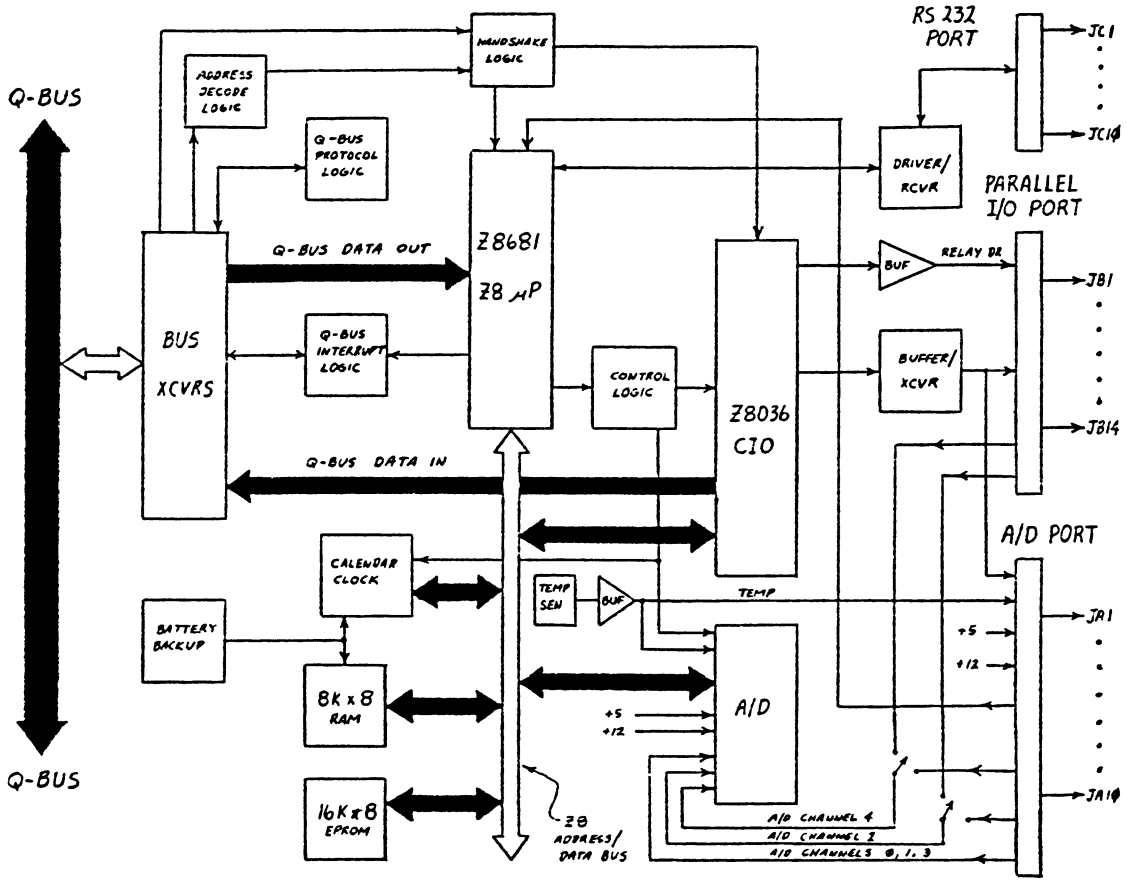
It is a good idea to verify that the other setup commands work also. When you are finished with setup mode, type <ESC> , the escape key, to return to QM_, at which point you can type P and "proceed" with where you were in the terminal emulation.

If your board has functioned correctly in the terminal emulation and setup modes, you have verified that the main hardware and software features are working.

BLANK

APPENDIX A

MODEL 170 Q-MONITOR
SYSTEM BLOCK DIAGRAM

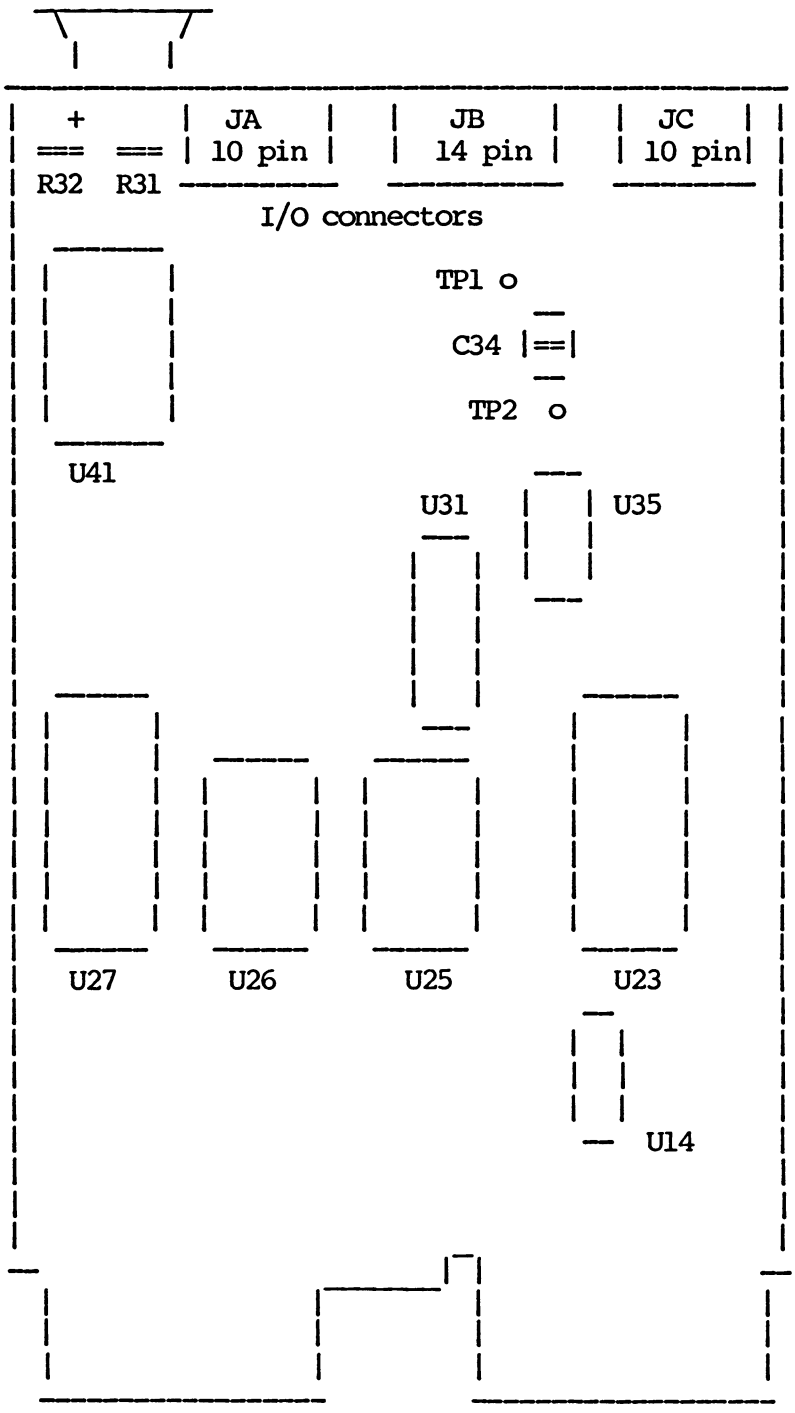


Note: When Q-Monitor is used in a redundant system configuration, A/D port becomes redundant system port.

BLANK

APPENDIX B

PHYSICAL LAYOUT



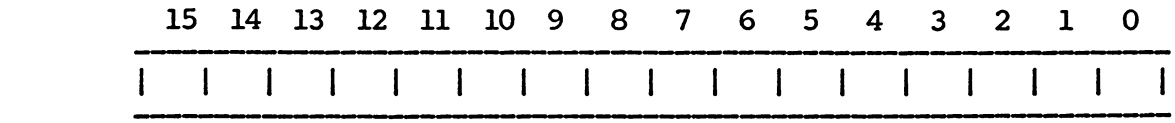
TP1	cal clock osc out
TP2	battery gnd (local gnd)
C34	calendar clock adj
R32	voltage ref
R21	voltage ref adj

U41	A/D converter
U35	calendar clock
U31	digital I/O buffer
U27	CIO
U26	CMOS RAM
U25	EPROM
U23	Z8
U14	Address decode PAL

BLANK

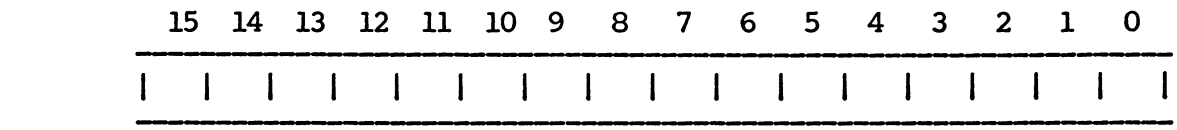
APPENDIX C

Q-BUS ADDRESSES AND BIT ASSIGNMENTS
FOR Q-MONITOR SERIAL INTERFACE REGISTERS
(shown for 18 bit system)



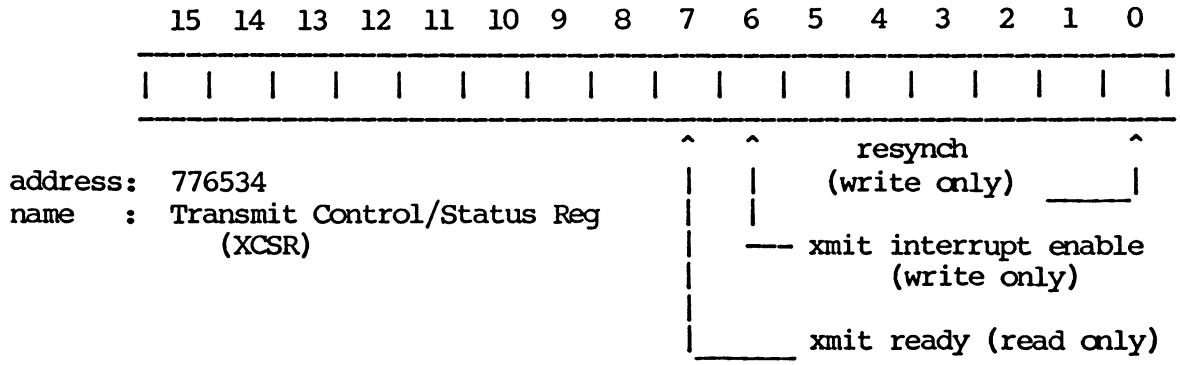
address: 776530
name : Receiver Control/Status Reg
(RCSR)

all other bits read as zero



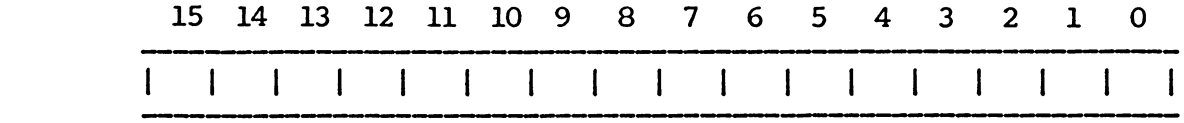
address: 776532
name : Receiver Buffer Register
(RBUF)

all other bits read as zero



address: 776534
name : Transmit Control/Status Reg
(XCSR)

all other bits read as zero



address: 776536
name : Transmit Buffer Register

all bits read as zero

BLANK

APPENDIX D

I/O PORT CONFIGURATION GUIDE

The Q-Monitor comes equipped with the 7-bit I/O port configured as 3 bits in, 4 bits out. The input bits and output bits are shown in the diagram on next page. A 74LS244 output buffer with tri-state outputs is the standard chip. The Q-Monitor uses one of the port lines on a permanent basis so that the user may employ only 7 of these lines at most.

There are three basic ways the port may be configured :

1) 3 bits are input, 4 bits are output. This is the standard arrangement.

2) The port may be 7 bits in or 7 bits out. The direction of the port is changed using host software. In this configuration, all outputs are tri-state.

3) The port may be 7 bits out. This configuration is allowed so that users needing open-collector outputs can get them. In this case, the user must provide his own pull-up resistors on the other end of his cable as the Q-Monitor doesn't have them. Choice (2) above cannot use open-collector transceivers because the Q-Monitor does not provide pull-up resistors on the board side of the transceivers.

NOTE: The standard Q-Monitor can only be configured as in 1) above, if the user desires another configuration, he must replace the port buffer chip (U31) as explained below. Any buffer change must be programmed as well, either by host or setup command. The relevant host command is SET PORT BUFFER and the relevant setup command is SET BUF.

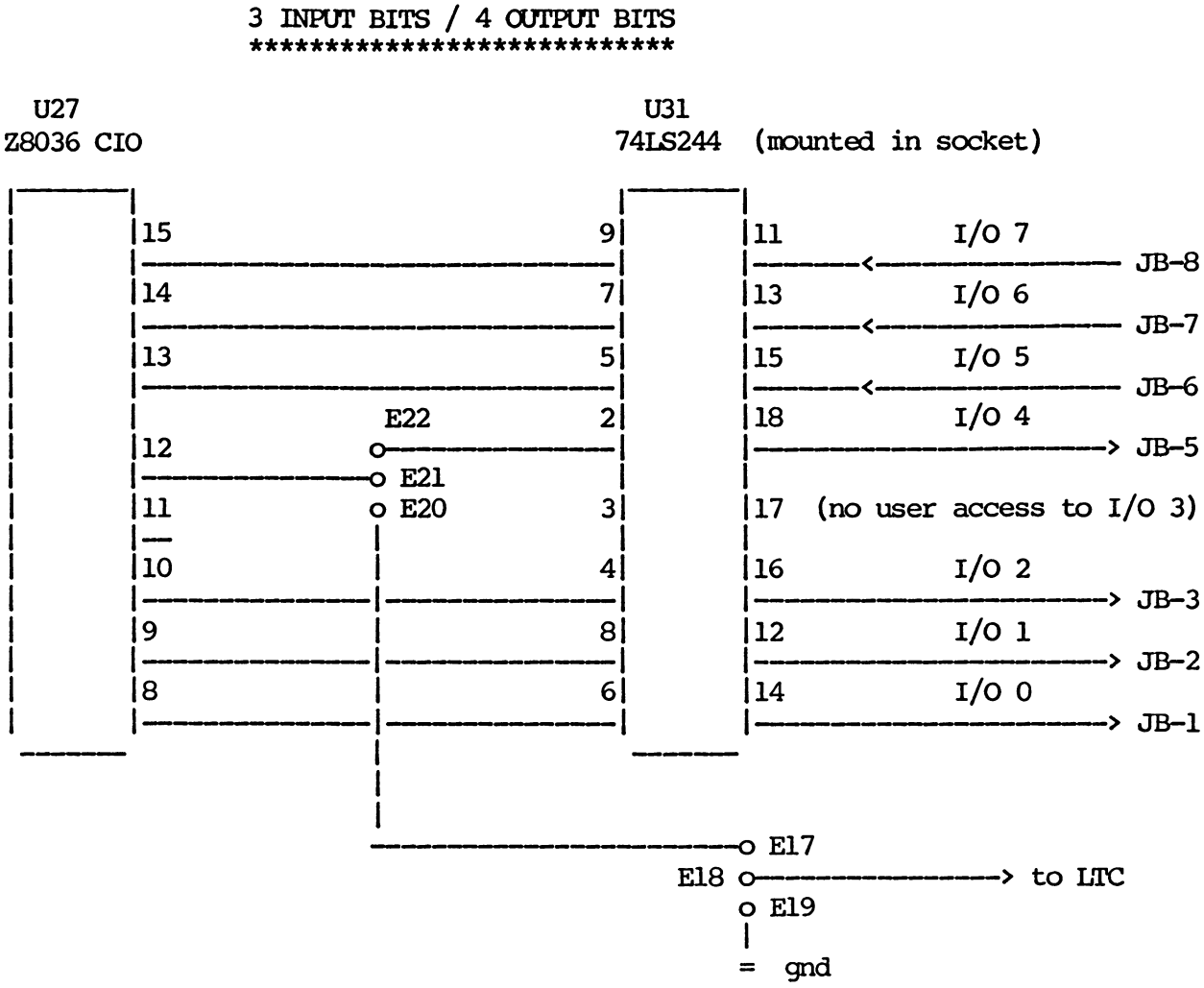
The table below shows the standard configuration, the optional configurations, some optional chips, and which optional chips may be ordered from Codar. Those chips available from Codar are marked with an asterisk (*). The list of useable chips is not complete, it is only a guide.

The polarity of the outputs or inputs is selectable using the host software. Any inversions of polarity are handled by the CIO. Thus it is not necessary to select any of the following chips just to get an inverted input or output.

Configuration	Some Useable Chips	Output type	Max Output Current, mA I(ol)	I(oh)
3 in, 4 out	74LS244	non-inverting 3-state outputs	24	-15
	74LS240	inverting 3-state outputs	24	-15
7 in / 7 out	74LS245 *	non-inverting 3-state outputs	24	-15
	74ALS640	inverting 3-state outputs	48	-15
	74ALS643	inv and non-inv 3-state outputs	48	-15
	74ALS645	non-inverting 3-state outputs	48	-15
7 out	74ALS642 *	inverting open-collector outputs	48	.1

note : on the optional parts available from Codar, I(ol)=24 mA max
to get I(ol)=48 mA , the 74ALSxxx-1 part must be obtained

The following diagram shows the I/O pin arrangement for the standard port configuration of 3 input bits/ 4 output bits



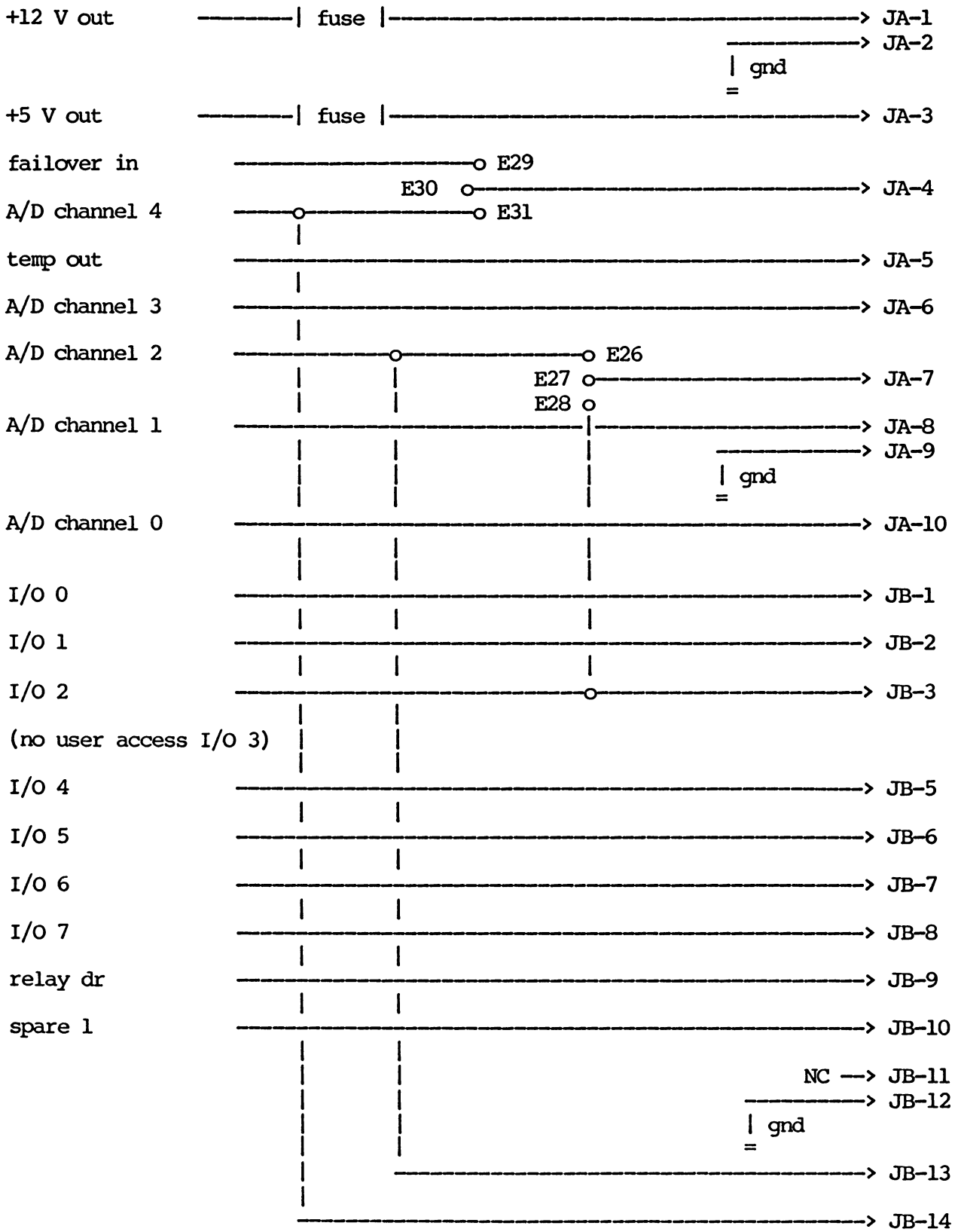
JUMPERING : If LTC option not used, jumper E22-E21 only
If LTC option used, jumper E21-E20, and E17-E18 only

If U31 is changed from standard chip (74LS244) to alter port configuration, see Appendix G for changes required to jumpers 23, 24, and 25.

BLANK

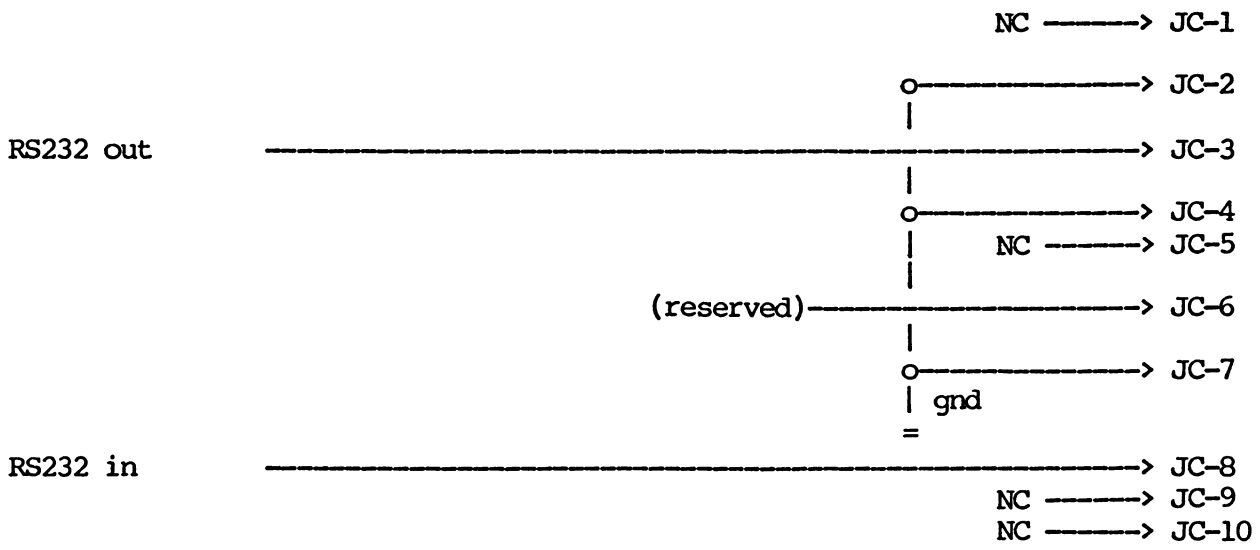
APPENDIX E

I/O CONNECTORS JA, JB, and JC



(continued on next page)

APPENDIX E (cont'd)



1) For redundant system operation, jumper E29-E30, and E27-E28. Note that in this configuration the user can still have access to A/D channels 2 and 4 via JB-13 and JB-14 respectively.

2) For redundant system operation, the cable mates to connector JA as shown below :

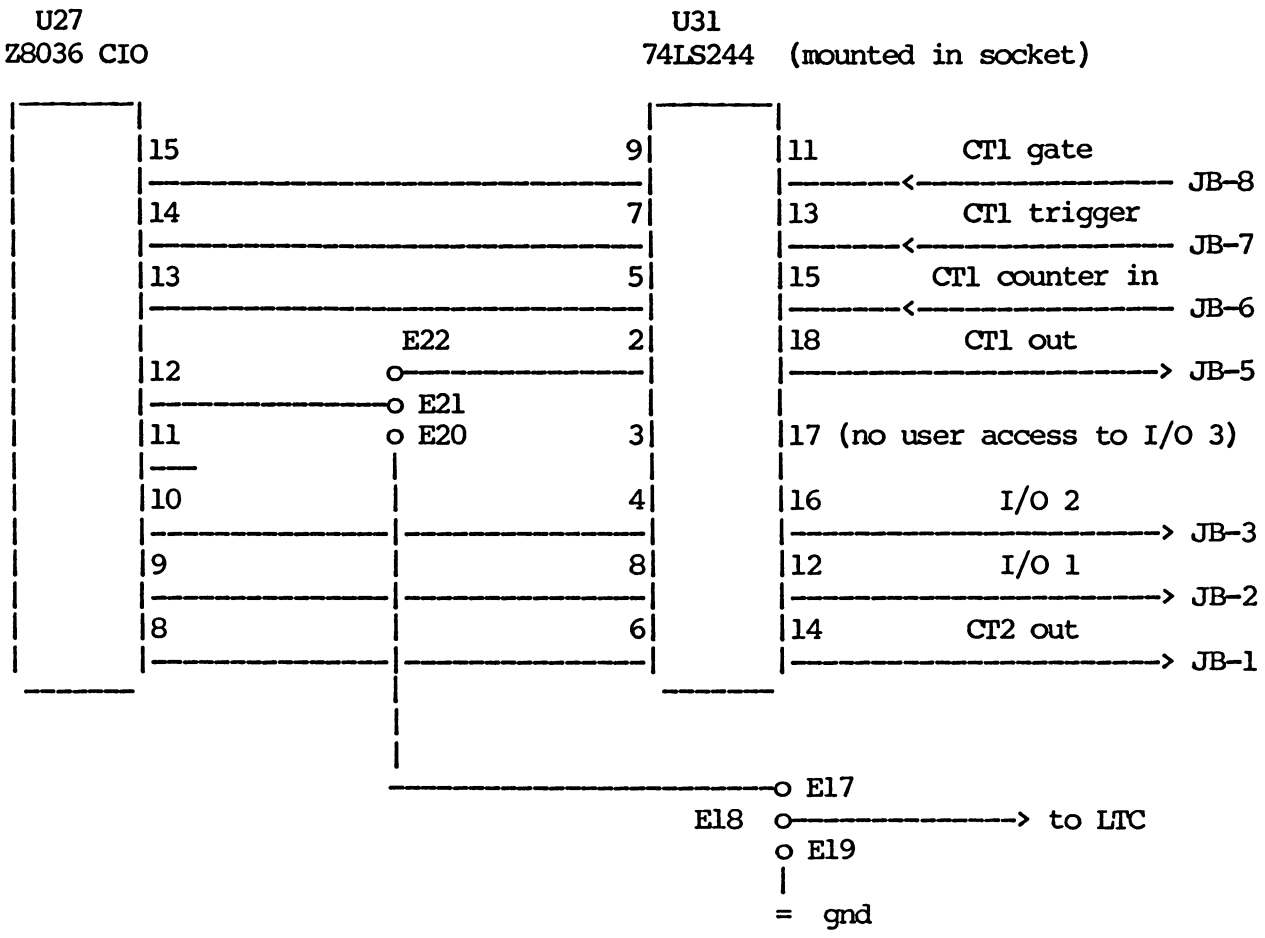
connector pin	signal name
JA-1	+12 V out
JA-2	gnd
JA-3	+5 V out
JA-4	failover in
JA-5	temperature sensor out
JA-6	temperature sensor in (A/D channel 3)
JA-7	failover out
JA-8	+5 V in
JA-9	gnd
JA-10	+12 V in

APPENDIX F

COUNTER/TIMER I/O GUIDE AND TIMING DIAGRAMS

The following diagram shows the I/O pin arrangement for the standard digital port configuration of 3 input bits, 4 output.

3 INPUTS / 4 OUTPUTS
****



NOTE : If CT1 out is not used for LTC, jumper E18-E19 only and jumper E21-E22 to connect CT1 out to JB

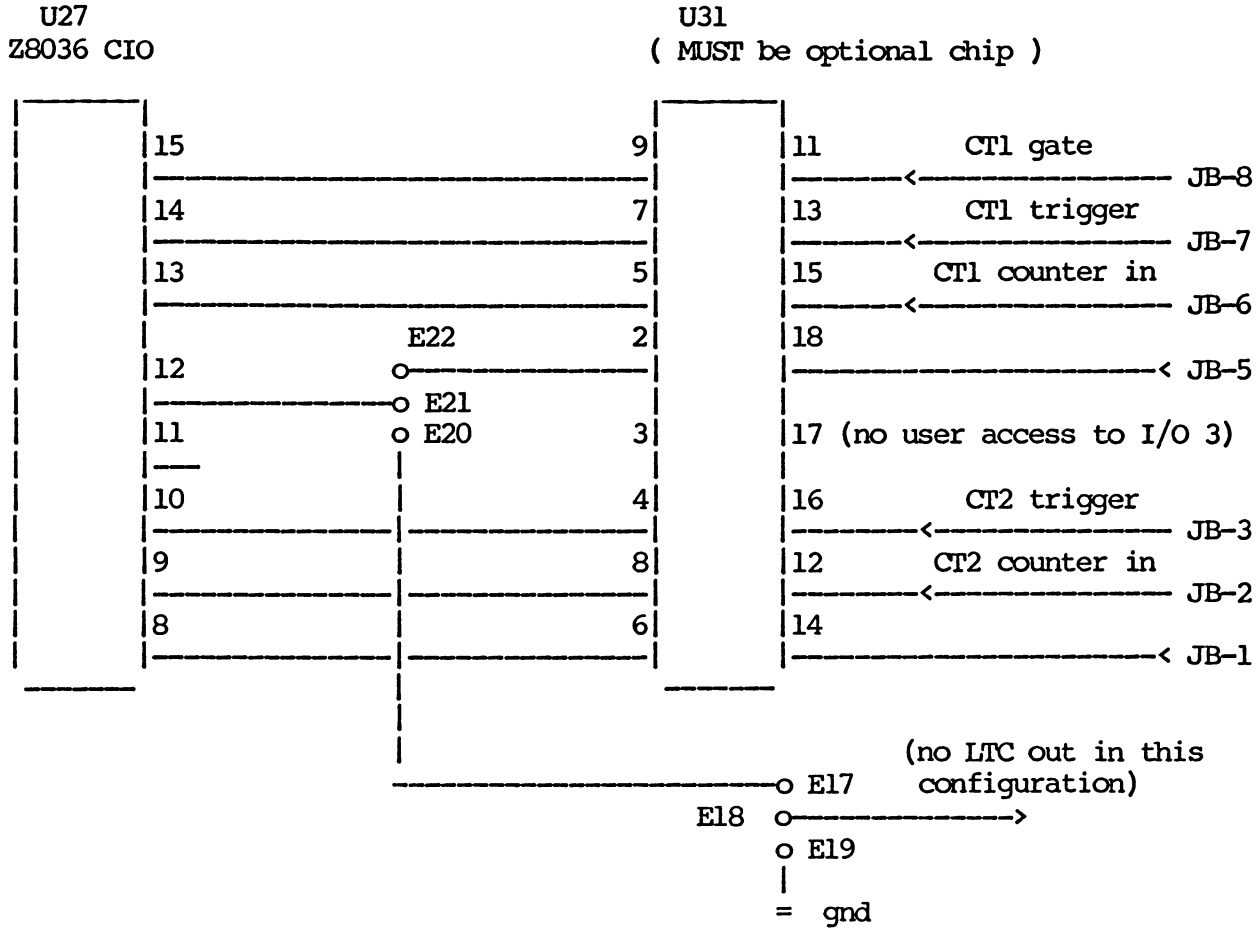
If CT1 out is used for LTC, jumper E17-E18 only and jumper E21-E20.

APPENDIX F (cont'd)

COUNTER/TIMER I/O GUIDE

The following diagram shows the I/O pin arrangement for the optional port configuration of 7 bits input.

7 INPUTS / 0 OUTPUTS
****

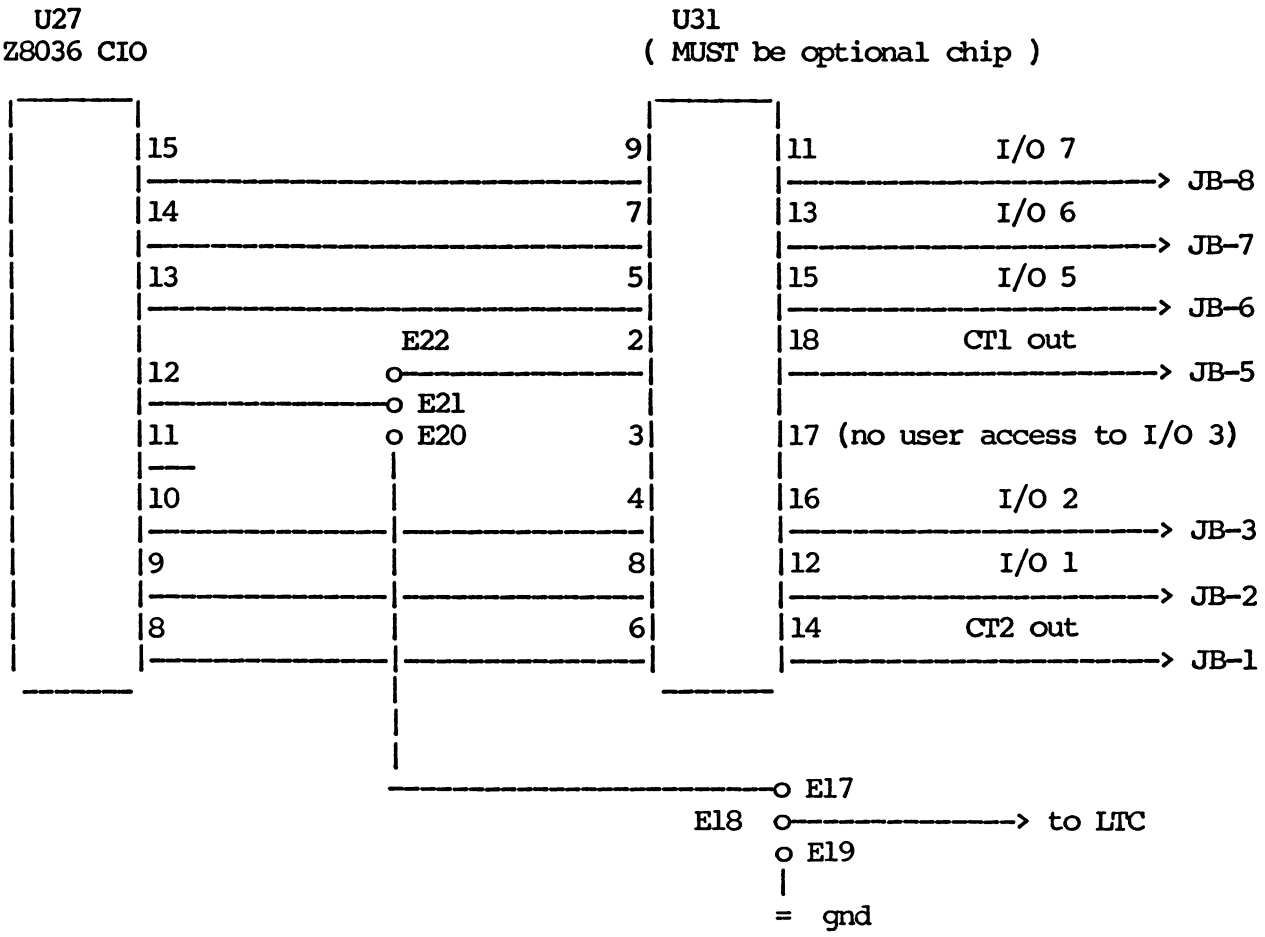


APPENDIX F (cont'd)

COUNTER/TIMER I/O GUIDE

The following diagram shows the I/O pin arrangement for the optional port configuration of 7 bits output.

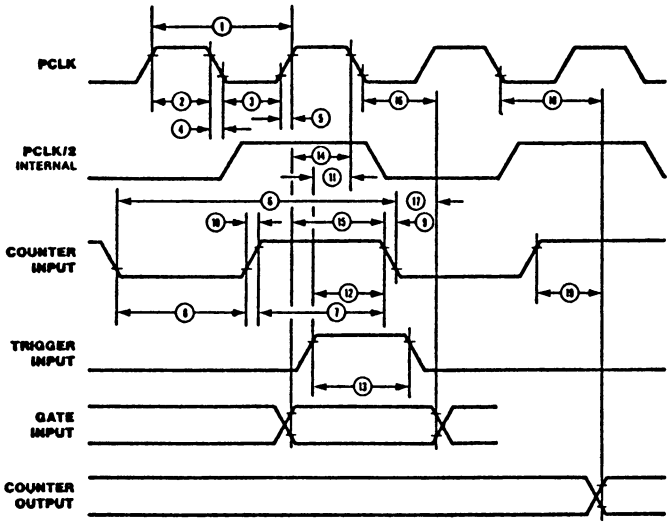
0 INPUTS / 7 OUTPUTS
****



NOTE: If CT1 out is not used for LTC, jumper E18-E19 only and jumper E21-E22 to connect CT1 out to JB

If CT1 out is used for LTC, jumper E17-E18 only and jumper E21-E20

COUNTER/TIMER TIMING



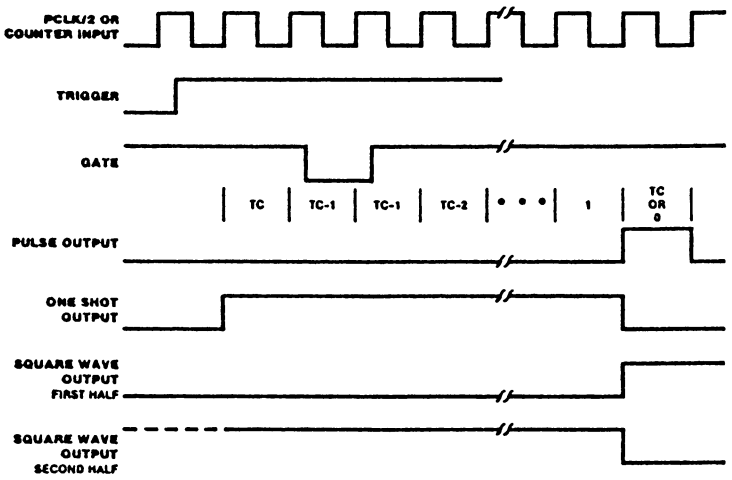
No.	Symbol	Parameter	Min	Max
1	TcPC	PCLK Cycle Time	250	4000
2	TwPCh	PCLK High Width	105	2000
3	TwPCl	PCLK Low Width	105	2000
4	TfPC	PCLK Fall Time		20
5	TrPC	PCLK Rise Time		20
6	TcCI	Counter Input Cycle Time	500	
7	TCIh	Counter Input High Width	230	
8	TwCIl	Counter Input Low Width	230	
9	TfCI	Counter Input Fall Time		20
10	TrCI	Counter Input Rise Time		20
11	TsTI(PC)	Trigger Input to PCLK ↓ Setup Time (Timer Mode)	150	
12	TsTI(CI)	Trigger Input to Counter Input ↓ Setup Time (Counter Mode)	150	
13	TwTI	Trigger Input Pulse Width (High or Low)	200	
14	TsGI(PC)	Gate Input to PCLK ↓ Setup Time (Timer Mode)	100	
15	TsGI(CI)	Gate Input to Counter Input ↓ Setup Time (Counter Mode)	100	
16	ThGI(PC)	Gate Input to PCLK ↓ Hold Time (Timer Mode)	100	
17	ThGI(CI)	Gate Input to Counter Input ↓ Hold Time (Counter Mode)	100	
18	TdPC(CO)	PCLK to Counter Output Delay (Timer Mode)		475
19	TdCI(CO)	Counter Input to Counter Output Delay (Counter Mode)		475

Notes :

1) PCLK is the same as the 500 KHz external clock
2) parameters 11, 12, 14, 15, 16, and 17 must be met to guarantee that trigger or gate are valid for the next counter/timer cycle.
3) all timing references assume 2.0 V for a logic "1" and 0.8 V for a logic "0".
4) all time units are in nanoseconds.

APPENDIX F (cont'd)

COUNTER/TIMING TIMING



Counter/Timer Timing Diagram

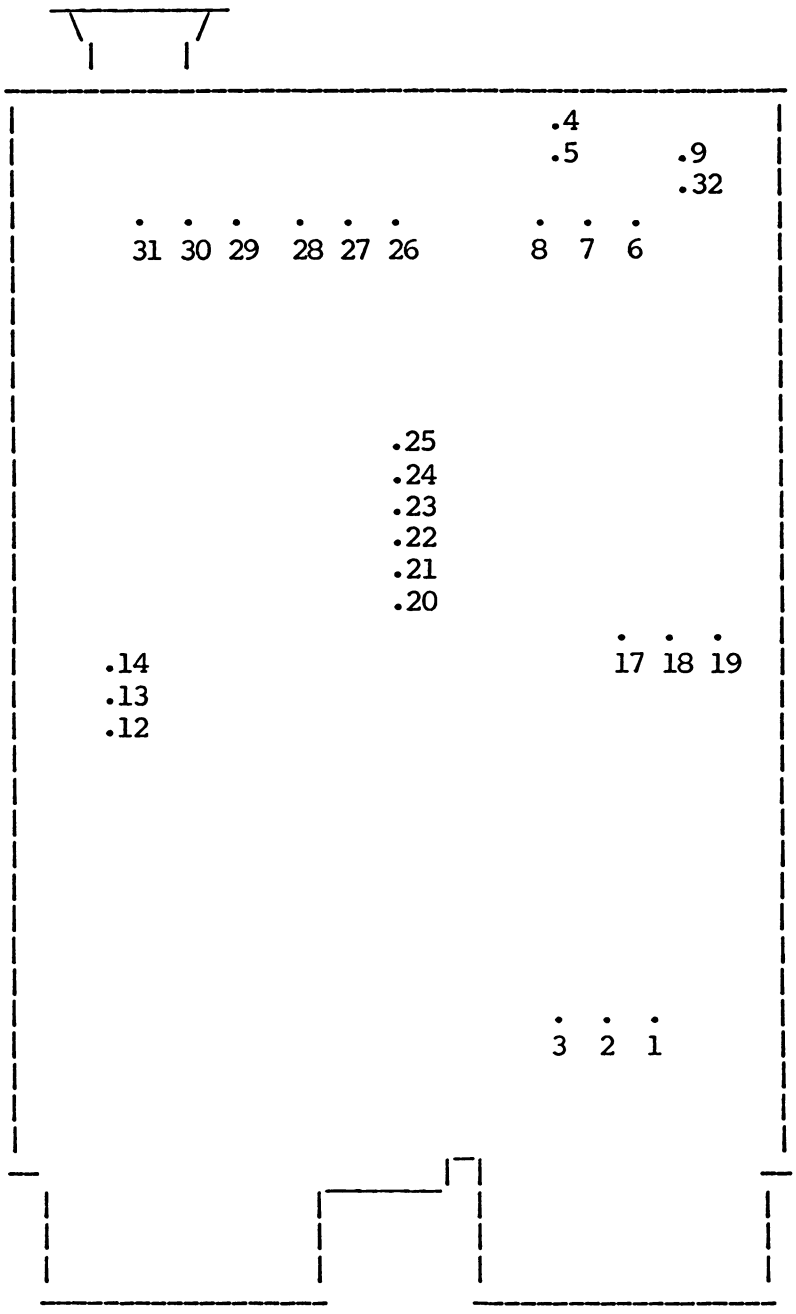
There are three duty cycles available for the timer/counter output: pulse, one-shot, and square wave. The diagram above shows the timing for the three waveforms.

When the pulse mode is specified, the output goes high for one cycle, beginning when the down-counter leaves the count of 1. In the one-shot mode, the output goes high when the counter/timer is triggered and goes low when the down-counter reaches 0. When the square-wave duty cycle is chosen, the counter/timer goes through two full sequences for each square-wave cycle. The initial trigger causes the down-counter to be loaded and the normal count-down sequence to begin. When a 1 count is detected on the down-counter's clocking edge, the output goes high and the time constant value is reloaded. On the clocking edge, when both the down-counter and the output are 1's, the output is forced low.

BLANK

APPENDIX G

JUMPER PIN IDENTIFICATION



note: some jumper pins are prefaced with "E" on silkscreen

Factory Installed Permanent Jumpers

a) Pins 1,2 come jumpered together from factory. For Codar use only.
b) Pins 4,5 must not be jumpered. For Codar use only.
c) Pin 7 jumpered to either 6 or 7 at factory. For Codar use only.
d) Pins 9 and 32 jumpered or not at factory. For Codar use only.

APPENDIX G (cont'd)

User Changeable Jumpers
Factory Configuration and User Options

e) Pin 13 - Pin 14 jumpered at factory for 8k x 8 RAM
Pin 13 - Pin 12 (remove 13-14) if 2k x 8 compatible RAM used

f) Pin 18 - Pin 19 jumpered at factory for no LTC configuration
Pin 18 - Pin 17 (remove 18-19) if Q-Monitor is to supply LTC

g) Pin 21 - Pin 22 jumpered at factory for no LTC configuration
Pin 21 - Pin 20 (remove 21-22) if Q-Monitor is to supply LTC

h) Pin 23 - Pin 24 jumpered at factory for digital I/O port set for 3 input bits, 4 out.
Pin 24 - Pin 25 (remove 23-24) for all other digital I/O port configurations

i) Pin 26 - Pin 27 jumpered at factory for external access to A/D converter channel 2 on connector JA
Pin 27 - Pin 28 (remove 26-27) if connector JA used for redundant system. A/D channel 2 still accessible on JB-13, see Appendix E.

j) Pin 30 - Pin 31 jumpered at factory for external access to A/D converter channel 4 on connector JA
Pin 29 - Pin 30 (remove 30-31) if connector JA used for redundant system port. A/D channel 4 is still accessible on JB-14.

APPENDIX H

TECHNICAL SPECIFICATIONS

Compatibility:	All Q-Bus processors, 16,18, or 22 bit
Module size :	5.2" x 8.9" dual wide
Q-Bus addresses:	776530 - 776536 standard
Q-Bus vectors:	330, 334 standard
Operating power:	5 V +/- 5%, 1.0 amps
Operating temp:	0 - 50 degrees C
Relative humidity:	5 - 95% noncondensing
Bus loading:	one LSI-11 bus load per line
Bus interface:	100% Q-Bus spec'd ICs
Battery back-up:	800 days without system power
Battery:	3.0 volt lithium, type 2430
Battery shelf life:	ten years
Calendar clock accuracy:	4 seconds/month typical, 8 seconds/month max.
A/D converter:	8-bit, muxed 8-channel, 5.35 V full scale, conversion time = 128 usecs. imput impedance = 2.5 megohms min
Temperature sensor:	0 - 70 degrees C
Digital I/O port:	8-bit, 74LS244 buffer standard
Relay/alarm output:	7406 open-collector
Serial Interface:	RS232 110 - 19200 baud, program selectable
EPROM memory:	27128 16k x 8
RAM memory:	CMOS SRAM, 6264-type 8k x 8
User accessible: counter/timers	two 16-bit or one 32-bit, maximum external clock frequency = 2.0 MHz

BLANK

APPENDIX J

WARRANTY

Codar Technology Inc. (hereinafter referred to as CTI), warrants the Q-Monitor to be free from defective material or workmanship for a period of 6 months from the date of shipment from Longmont, Colorado, USA. This warranty applies to the original purchaser only and may not be transferred without the written permission of CTI. CTI under this warranty is limited to repairing or replacing the defective product components which, in CTI's judgment, shall have been originally defective or to have failed in normal operation during the warranty period. Disposable items, such as batteries, fuses, etc., are not covered by this warranty.

The CTI equipment must be returned to the factory (Longmont, Colorado, USA) with a return material authorization (RMA) number obtained via telephone from CTI, within 6 months from the date of original shipment from Longmont, Colorado, USA. The purchaser shall prepay shipping charges to CTI and CTI shall pay shipping charges to return the product to purchaser. However, purchaser shall pay all shipping charges, duties, and taxes for equipment returned to CTI from another country.

CTI warrants that its software and firmware designated by CTI for use with or on a piece of equipment, will execute its programming instructions when properly installed on that piece of equipment. CTI does not warrant that the operation of the piece of equipment, or software, or firmware will be uninterrupted or completely error free.

This warranty shall not apply to any damage or defect caused by injuries received in shipment, or any defects or failure resulting from accident, misuse, improper or inadequate maintenance by purchaser, purchaser-supplied software or firmware or interfacing, modifications or connections or installations or adjustments contrary to instructions furnished by CTI in writing, unauthorized repairs, operation outside of the environmental specifications for the equipment, or exposure to corrosive environments.

This warranty does not apply to, nor does CTI assume any liability for, the use of the equipment in any life support application.

CTI shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

No other warranty is expressed or implied. CTI specifically disclaims the implied warranties of merchantability and fitness for a particular purpose.

This warranty is in lieu of all other representations or warranties expressed or implied and no agent or representative of CTI is authorized to assume any other obligations in connection with the sale and purchase of this equipment.

BLANK

APPENDIX K
INDEX I

A/D converter 3-1, 3-5, 3-6, 4-2, 4-3
addresses, Q-Bus 3-1, 4-2
addresses, Q-Bus, alternate 3-2
alarm 3-4

battery backup 3-3, 3-5, 3-8
baud rate 3-3, 4-1
baud rate, set 2-3
baud rate, show 2-4
boot 3-2
break key 2-1, 3-2, 3-12, 4-4
buffer, I/O, set 2-3
buffer, I/O, show 2-4

C-D backplane 4-1
cabling, board 3-7, 3-11
calendar clock 3-5, 4-3
CIO chip 3-3, 3-4
CMOS RAM 3-1
commands, board 2-1
configuration, I/O port 3-9
connectors, I/O 4-2
console terminal 4-2
counter input 3-10
counter/timer 3-1, 3-3, 3-8

data log 3-3
date/time, set 2-2
date/time, show 2-4
definitions 1-1
digital I/O 1-1
DMA 3-2

escape key 2-1
external inputs, A/D 3-6
external inputs, counter 3-10

failover, redundant system 3-7, 3-9

gate input, counter 3-10

halt 3-2

host system 1-1, 2-1

I/O port 1-1, 3-1, 3-3, 3-4, 3-9, 4-2, 4-3
installation 4-1
interrupt priority, set 2-3
interrupt priority, show 2-4
interrupt vectors, set 2-3
interrupt vectors, show 2-4
interrupts 3-2, 4-1

jumpers 4-2

LTC 3-9, 4-2

memory 3-3
MicroVax 4-1

oscillator, set 2-4
output buffer 4-3

parallel I/O 1-1
parameters monitored 1-2
parity, serial line unit 4-1
parity, set 2-2
parity, show 2-4
Port B 3-4
powerup 4-4
powerup message 2-1
programming Q-Monitor 1-2

Q-Bus, compatibility 4-1
Q-Bus, interface 3-1
QMsetup_ 2-1, 4-4
QM_ 2-1, 4-4

reboot 3-2
recall factory, command 2-4
recall saved, command 2-5
redundant system 1-2, 3-5, 3-6, 3-9, 4-2
registers, Q-Bus 3-1
relay output 3-4
reply terminator, set 2-3
reply terminator, show 2-4
restrictions, set commands 2-4
RS232 port 1-1

save current, command 2-5
serial interface 3-1
serial port 1-1, 2-1, 3-11, 4-2
setup commands 1-2, 2-1, 4-4
system compatibility 4-1

temperature sensor 3-1, 3-7
terminal emulation 2-1
terminal usage 2-1, 4-1, 4-4
thresholds, user 1-2

trigger input, counter 3-10

UART 3-11
UPS system 3-5

vectors, Q-Bus interrupt 3-2, 4-1

watchdog timer 3-3, 3-4

Z8 microprocessor 3-1, 3-2

BLANK

NOTE :

APPENDIX PAGES F-4 AND F-5 ARE REPRINTED FROM THE ZILOG Z8036 COUNTER/TIMER and PARALLEL I/O UNIT TECHNICAL MANUAL

BLANK

Separator

Separator

TABLE OF CONTENTS, PART II

1.0 Introduction
- 1.1 Event Monitoring 1-2
- 1.2 Calendar Clock 1-2
- 1.3 Watchdog Timers 1-3
- 1.4 Counter/Timers 1-3
- 1.5 A/D converter 1-4
- 1.6 Digital I/O 1-4
- 1.7 Hobbs Meters 1-4
- 1.8 Non-Volatile RAM 1-4
- 1.9 Terminal Emulation 1-5
- 1.10 Keyboard Commands 1-6

2.0 Event Monitoring
- 2.1 The Event Table 2-2
- 2.2 Q-Monitor Events and Actions 2-2
  - 2.2.1 Events 2-3
    - 2.2.1.1 Reboots (warm starts) 2-4
    - 2.2.1.2 Powerups (cold starts) 2-4
    - 2.2.1.3 Watchdog A Timeouts 2-4
    - 2.2.1.4 Watchdog B Timeouts 2-4
    - 2.2.1.5 UPS Transitions 2-4
    - 2.2.1.6 A/D Excursions 2-4
    - 2.2.1.7 Digital I/O 2-4
    - 2.2.1.8 Counter/Timer 1 2-5
    - 2.2.1.9 Counter/Timer 2 2-5
    - 2.2.1.10 Failover Input 2-5
  - 2.2.2 Actions 2-5
    - 2.2.2.1 Counting Events 2-5
    - 2.2.2.2 Logging Events 2-5
    - 2.2.2.3 Interrupt 2-6
    - 2.2.2.4 Reboot 2-6
    - 2.2.2.5 Alarm 2-6
    - 2.2.2.6 Transmit Message 2-6
    - 2.2.2.7 Failover 2-6
- 2.3 Examining Logs 2-6
  - 2.3.1 SHOW TABLE 2-6
  - 2.3.2 SHOW RECORD 2-7

3.0 Calendar Clock

4.0 Watchdog Timers

5.0 Counter/Timers

6.0 A/D Converter

7.0 Digital I/O
7.1 Port Data 7-1
7.2 Port Configuration 7-1
7.2.1 Data Direction 7-2
7.2.2 Path Polarity 7-2
7.2.3 Special I/O 7-2
7.2.4 Pattern Matching 7-2
7.3 Buffer Type 7-3
7.4 Port Protection 7-3

8.0 Hobbs Meters

9.0 Non-Volatile RAM

10.0 Terminal Emulation
10.1 Registers and Vectors 10-1
10.2 Baud Rates 10-2
10.3 I/O Driver Considerations 10-3
10.3.1 RT11 10-3
10.3.2 RSX Variants 10-3
10.3.3 MicroVMS 10-4
10.3.4 ELAN 10-4

11.0 Keyboard Commands
11.1 QMsetup 11-2
11.3 QMdebug 11-2

12.0 Theory of Operation
12.1 Terminal Emulation and Communications 12-1
12.2 Event Interrupts 12-2
12.3 Non-Volatile Storage 12-2
12.3.1 Dates and Times 12-2
12.3.2 Parameter Page 12-3
12.3.3 Event Table 12-3
12.3.4 Event Logs 12-4
12.3.5 User Data 12-4

13.0 Programming the Q-Monitor

14.0 Command Descriptions

Appendix A Command Summaries
A.1 Commands Ordered by ASCII Command Code A-1
A.2 Commands Ordered Alphabetically by Mnemonic A-6
A.3 Commands Ordered by Functional Group A-9
A.3.1 Startup Greeting A-9
A.3.2 Configuration A-9
A.3.3 Communications and Terminal Emulation A-9
A.3.4 Timekeeping A-10
A.3.5 Watchdog Timers A-10
A.3.6 Counter/Timers A-10
A.3.7 A/D Converter A-11
A.3.8 Digital I/O Port A-11
A.3.9 Event Monitoring A-11
A.3.10 UPS and Failover A-12
A.3.11 Interrupt Control A-12
A.3.12 Miscellaneous A-12

Appendix B Factory Defaults

Appendix C Support Kit

Appendix D Example Fortran Program

Appendix E 7-Bit ASCII Codes

Appendix F Index, Part II

BLANK

INTRODUCTION

CHAPTER 1

The Q-Monitor is a microprocessor based multiple function utility board compatible with all 16-, 18-, and 22-bit Q-Bus systems. On the single "dual-wide" board are a CMOS calendar clock, an A/D converter, several counter/timers, ROM, CMOS RAM, serial and parallel I/O hardware, a Lithium battery capable of operating the CMOS devices for several years without external power, and connectors which provide external access to the Q-Monitor's inputs and outputs.

A central feature of the Q-Monitor is that it emulates a Digital DLV11 serial port: any RS232-C terminal can be connected to the Q-Monitor, and normal terminal I/O can be performed between that terminal and any host system whose terminal driver supports the DLV11.

More importantly, the Q-Monitor's sophisticated system monitoring capabilities can be activated and controlled very easily by programs in the host system which transmit special escape sequences to the Q-Monitor. These escape sequences are not sent to the terminal, but rather are interpreted as commands for the Q-Monitor itself.

NOTE

> Even if your system's terminal driver doesn't support the DLV11, the Q-Monitor can still be controlled easily via direct access to its registers; this topic is discussed in more detail elsewhere in this manual, and the Support Kit contains programs which use this technique on RT11, RSX and MicroVMS.

Briefly, the Q-Monitor provides:

- o "Event" monitoring and logging
- o A calendar clock with leap year logic
- o A two stage watchdog timer
- o Two programmable counter/timers
- o An 8-bit 8-channel A/D converter with programmable thresholds
- o Digital I/O with programmable pattern recognition
- o "Hobbs meter" recording of uptime and downtime
- o Non-volatile RAM for data storage
- o DLV11 terminal emulation with programmable baud rates
- o Self-contained keyboard commands

All Q-Monitor behavior is controlled by two internal non-volatile data structures, known as the Parameter Page and the Event Table. The Event Table controls Event Monitoring; the Parameter Page controls all other features.

Both the Event Table and the Parameter Page can be altered by transmitting escape sequences (commands) to the Q-Monitor; certain items can also be changed from an on-board command processor called "QMsetup" (see below).

1.1 Event Monitoring

The Q-Monitor can be conditioned to respond to these "Events":

- o Host reboots (warm starts)
- o System powerups (cold starts)
- o Watchdog A timeouts
- o Watchdog B timeouts
- o UPS system transitions
- o A/D excursions
- o Digital I/O pattern matches
- o Counter/Timer 1 reaching terminal count
- o Counter/Timer 2 reaching terminal count
- o Failover Input transitions

The Q-Monitor can also exchange information on the status of its host system with a Q-Monitor installed in another system (the feature referred to as "Failover"), and it can track transitions of an Uninterruptible Power Supply and other external conditions.

With a few exceptions, it can be directed to take one or more of the following Actions in response to any Event:

- o ignore the Event (NOP)
- o increment the Event's counter
- o log the Event
- o interrupt the host
- o reboot the host
- o assert the Alarm relay
- o transmit a message to the terminal
- o assert Failover Output

NOTE

A future Q-Monitor release will support transmission of messages to a remote terminal via an autodial modem.

"Logging" an Event consists of recording the date and time of the event, the Action that was taken, and any specific information appropriate for the particular event type. The Q-Monitor maintains ten Event Logs (one for each Event type), which can be read by host programs.

In this way, the Q-Monitor provides an extensive history of the operation of its host system. This can be a valuable tool in system maintenance and optimization.

1.2 Calendar Clock

The Q-Monitor provides date and time to the host as eight numeric values:

Year	0-99
Month	1-12
Date	1-28, 29, 30, or 31 as appropriate
Hour	0-23
Minute	0-59
Second	0-59
Tenth	0-9
Day	1-7 for Monday - Sunday

The Q-Monitor updates these values internally every 100 msec; they can be read by the host at any time via the SHOW CLOCK command, which encodes the eight bytes and returns them as a string of 16 ASCII hexadecimal characters; the SET CLOCK command is used to adjust the calendar clock.

QMsetup also provides SET and SHOW CLOCK commands, for time maintenance directly from the Q-Monitor's terminal.

1.3 Watchdog Timers

The Q-Monitor provides two watchdog timers (only one of which is active at any time): Watchdog A is active immediately after a powerup; and Watchdog B, which is activated after Watchdog A terminates and remains active until the system is powered down.

Distinct time constants and Action codes (including "NOP") can be specified for Watchdogs A and B, allowing the Q-Monitor to differentiate between system failures which occur during startup and those which occur during subsequent operations.

The active Watchdog (A or B) is a 16-bit down counter, decremented at a 10 Hz rate, providing timeout intervals from 0.1 second to 1:49:13.6. If the counter reaches 0, the Q-Monitor will automatically perform any selected Action without further host processor intervention. The active watchdog timer can be reset either by issuing the RESET WATCHDOG command, or by reading the Q-Monitor's transmit data buffer (TBUF = RCSR + 6).

1.4 Counter/Timers

Two programmable counter/timers are available for counting events, timing intervals and generating waveforms. These may be used independently or in one of three linked modes. External access to gate, count and trigger lines is available.

Either or both may be configured to generate any of the Q-Monitor's Actions, including host interrupts.

NOTE

Counter/Timer 1 can be configured to drive the Q-Bus BEVNT line, providing a programmable "line time" clock for use by the host's operating system.

1.5 A/D converter

The Q-Monitor round-robin samples its on-board 8-bit, 8-channel A/D converter at 100 Hz: each channel is sampled once every 80 msec. Independent upper and lower thresholds may be set for each channel, and the Q-Monitor can be directed to generate any of its Actions, including host interrupts, when any threshold is exceeded. Every 100 msec the Q-Monitor scans the data to update the maximum and minimum for each channel and to detect any threshold excursions.

Channels 5-7 are reserved for monitoring the backplane's +12V and +5V supplies, and the Q-Monitor's on-board temperature sensor; channels 0-4 are available for connection to your own signals, unless you are are using the Failover facility. In that case channels 0, 1, and 3 are used for the remote systems's +12V, +5V and temperature, respectively. All user inputs must be 0-5.35V.

1.6 Digital I/O

The Q-Monitor's digital I/O port can be used to drive or monitor up to seven external TTL signals. These seven lines can be seven inputs, seven outputs, or three inputs and four outputs, depending upon the buffer chip installed at U31.

Pattern recognition (sensitivity to levels or transitions) can be programmed independently for each line, and the Q-Monitor can be conditioned to generate any Action when a pattern match occurs.

1.7 Hobbs Meters

The Q-Monitor accumulates a record of uptime and downtime (in units of .01 hour) since the battery was installed. These are zeroed by the SET BATTERY command, and can be read with the SHOW UPTIME and SHOW DOWNTIME commands.

1.8 Non-Volatile RAM

Portions of the Q-Monitor's RAM that are not used for other functions are available for storing host system data.

1.9 Terminal Emulation

To the host processor, the Q-Monitor appears to be a single line RS232-C serial interface (e.g. DLV11); either polled I/O or interrupts may be used to access it. Communication with the terminal attached to the Q-Monitor is "transparent", in that all output (other than Q-Monitor commands) goes to the terminal without modification, and all input is passed to the host as normal keyboard data. As shipped, parameter settings are:

Receiver control and status register (RCSR)	176530 octal
Vector	330 octal
Q-Bus priority	4
Baud rate	110
Data bits/Parity	Eight/Disabled
100 cps limit	Enabled
BREAK recognition	Enabled
Stutter	Enabled

All of these parameters except RCSR are programmable; RCSR can be changed by swapping the Q-Monitor's socketed address PAL.

The "100 CPS Limit" and "Stutter" are designed to accomodate host system terminal drivers that cannot handle high speed input: the 100 cps limit, if enabled, restricts the Q-Monitor's data input rate; "Stutter", if enabled, causes a 10 to 20 msec hesitation before replying to a command from the host.

By changing the interrupt vector from 330 to 60 and substituting the 177560 address PAL, the Q-Monitor can be installed as the console terminal.

As shipped, all other Q-Monitor features are available but inactive.

NOTE

1) The Q-Monitor is fully compatible with the RSX and RT11 operating systems, including operation as the console terminal. However, when installed as the console terminal there is one restriction to ODT operation: having entered ODT from the operating system, it is not possible to resume execution without rebooting. This restriction will be removed as soon as practicable.
2) Although DLV11 interfaces are supported under all LSI-11 operating systems and under VAX ELAN, Digital has chosen not to support them under MicroVMS. All of the Q-Monitor's features can be exercised under MicroVMS, but other access techniques must be employed (see Support Kit).

1.10 Keyboard Commands

In normal operation the Q-Monitor is commanded and controlled through escape sequences transmitted by programs in the host system, while the terminal connected to the Q-Monitor will seem to be connected directly to the host. However, you may use the terminal to communicate with the Q-Monitor itself. If the Q-Monitor's BREAK recognition is enabled, typing "BREAK" suspends all normal Q-Monitor operations and lets you enter one of the on-board command processors:

QM_ to halt or reboot the host processor, or to enter

QMsetup_ to change Q-Monitor configuration parameters, or

QMdebug_ a Q-Monitor debugging and maintenance facility

QMsetup lets you modify certain of the Q-Monitor's operational parameters directly from the terminal, independently of programs in the host system. A limited subset of commands, sufficient to allow installation of the Q-Monitor on your system, is currently implemented; future Q-Monitor releases will include an expanded set of commands in QMsetup.

If necessary, the SET BREAK command can be used to disable BREAK recognition to prevent tampering.

EVENT MONITORING

CHAPTER 2

The Q-Monitor recognizes ten "Events", responding to them by recording the date and time of their occurences and/or by taking certain other "Actions" that you can specify. For each of the Events an Event Log containing a number records is maintained. Recording the Events in the Event Log is referred to as "logging", and is one of the Actions.

This monitoring of Events is controlled by the Event Table, a non-volatile data structure maintained in the Q-Monitor's CMOS RAM. You can tailor the Event Table to suit your local requirements by using the commands listed in Appendix A, section A.3.9.

With these commands, you can define which Actions will be taken in response to each Event, how many records to keep in each Event Log, etc. The Q-Monitor's Events and Actions are described in detail in the next several sections, but before proceeding, let's look at two cases where the similarity between the Event and the Action may be confusing:

Reboot: When an external reboot occurs (e.g. someone hits the "Restart" switch), the Q-Monitor notices this Event, and may take some Action; however,

If "Reboot" is specified as an Action for some Event (e.g. for Watchdog B timeout), the Q-Monitor will cause a system reboot by toggling BDCOK when the Event occurs.

Failover: If the Failover Input line changes state (indicating that a redundant system has gone down or come up), the Q-Monitor may take some Action to notify the local system of the situation; but,

If the local system experiences a failure (e.g. temperature exceeds the threshold), the Q-Monitor may assert Failover Output to notify a backup system.

2.1 The Event Table

The Event Table contains an entry for each of ten Event types; each entry, in turn contains several fields which control the Q-Monitor's response to that specific Event.

Two commands are used to determine the contents of the Event Table:

SHOW ACTIONS	returns the Action code for each Event
SHOW TABLE	returns the values of all fields (other than the Action code) for each Event

Three commands define or reinitialize these fields:

SET TABLE	defines the logging mode and the total number of records in each Log
SET ACTIONS	defines the Action codes for all Events
RESET LOG	zeroes the Event counter and the number of records in use in the specified Log

The following table summarizes the fields of the Event Table, and the commands that can be used to examine and define them:

Field	Function	Write with	Read with
Action code	defines which Action the Q-Monitor will take in response to this Event	SET ACTIONS	SHOW ACTIONS
Event counter	records the number of times this Event has occurred since this Log was last cleared	RESET LOG (zeroes the counter)	SHOW TABLE
Total number of records	defines the size of this Log, and also whether "Purge mode" or "Stop mode" is used	SET TABLE	SHOW TABLE
Record length	defines the number of bytes in each record of this Log	cannot be changed	SHOW TABLE
Number of records filled	shows the number of records in this Log that are currently in use	RESET LOG (zeroes the field)	SHOW TABLE

2.2 Q-Monitor Events and Actions

The Events recognized by the Q-Monitor are:

Event Code	Event
1	System reboots (warm starts)
2	Powerups (cold starts)
3	Watchdog A timeouts (startup timeouts)
4	Watchdog B timeouts ("normal" timeouts)
5	UPS transitions (switchover to battery power)
6	A/D excursions (A/D channel exceeds a threshold)
7	Digital I/O line conditions (transitions or levels)
8	Counter/Timer 1 reaching terminal count
9	Counter/Timer 2 reaching terminal count
10	"Failover" input transitions

For each of these Events, the Q-Monitor can be directed to:

Action Code	Action
0	ignore the Event
1	increment the Event counter
2	log the Event
4	interrupt the host processor
8	reboot the host processor
16	set an alarm signal
32	transmit a message to the terminal (NOT YET IMPLEMENTED)
64	assert the "Failover" output

NOTE

- Action codes can be added to yield multiple actions; e.g. an Action code of 6 specifies that the Event should be logged, and generate an interrupt. However, some common sense rules should be observed ...
- "Interrupt" is not a valid Action for Reboots, Powerups, and Watchdog A timeouts.
- "Reboot" should not be selected for Powerups, Reboots, UPS transitions, or Failover Inputs
- "Failover" should not be selected for Failover Inputs

2.2.1 Events

2.2.1.1 Reboots (warm starts) - Q-Bus BDCOK (the "Restart" line) is toggled by a source other than the Q-Monitor itself.

2.2.1.2 Powerups (cold starts) - A complete power up cycle occurs.

2.2.1.3 Watchdog A Timeouts - The first stage watchdog timer reaches its terminal count before any program in the host system issues the RESET WATCHDOG command (or reads TBUF).

2.2.1.4 Watchdog B Timeouts - The second stage watchdog timer reaches its terminal count before any program in the host system issues the RESET WATCHDOG command (or reads TBUF).

2.2.1.5 UPS Transitions - The UPS input line is asserted by an Uninterruptible Power Supply (battery backup system) indicating that it has switched from line power to battery power.

2.2.1.6 A/D Excursions - Any A/D converter channel exceeds either its lower or upper threshold.

NOTE

The A/D data are examined every 100 msec (10 Hz) to determine which, if any, of the channels are out of tolerance. For EACH channel which exceeds either its upper or lower threshold, the specified Action(s) will be taken. The Action(s) will continue until either the value is once again within the thresholds, or the Action code is changed. The Action code that is specified for A/D excursions applies to all A/D channels -- if the Action is "Interrupt" and 3 channels are outside their thresholds, the host system will receive 3 interrupts through the A/D vector every 100 msec until conditions change.

2.2.1.7 Digital I/O - A pattern match occurred on any of the digital I/O port lines.

2.2.1.8 Counter/Timer 1 - CT1 reached terminal count (zero)

2.2.1.9 Counter/Timer 2 - CT2 reached terminal count (zero)

2.2.1.10 Failover Input - A transition occurred indicating that a redundant system went down or came up. Failover Input can be either active high or active low (the default); it is debounced by sampling at 10 Hz, and latched so that its Action is generated once for each false-to-true transition and once for each true-to-false transition.

2.2.2 Actions

2.2.2.1 Counting Events - If bit 0 of the Action code is set when an Event occurs, the 16-bit counter specific to that Event type is incremented. The values for all Event counters are returned by the SHOW TABLE command.

2.2.2.2 Logging Events - If bit 1 of the Action code is set when an Event occurs, the Event is logged: the date and time of its occurence and the Action code in effect at that time are recorded. For some Event types, one or more bytes of special data are recorded.

The information thus accumulated is stored in one of ten Event Logs, each of which consists of 0 to 127 records. The records in these Logs may be examined using the SHOW RECORD command. Each Log may have a different number of records, but the number of records in the Logs must be specified for all Logs at once using the SET TABLE command. The SET TABLE command clears all previously logged information.

The behavior of the Q-Monitor when the last available record of a Log is filled can also be controlled independently for each Log. Two options are available:

- "Purge Mode" logging (the default) -- the oldest Log record (1) is discarded each time a new instance of the Event occurs, or
- "Stop Mode" logging -- when the Log fills, further Events are not recorded (however, they continue to be counted and to generate any Action specified by the Action code)

Purge mode for a Log is enabled when you set the number of records in that Log to a positive value; Stop mode is enabled if you specify a negative value (see SET TABLE).

2.2.2.3 Interrupt - The Q-Monitor interrupts the host through a vector specific to the Event. For digital I/O transitions, RBUF contains the contents of the 8-bit data register and MUST be read; for A/D excursions, RBUF contains the channel number on which the excursion occurred and MUST be read; for other Events RBUF is undefined and MUST NOT be read.

NOTE

When the Q-Monitor is configured to generate repetitive Event interrupts (e.g. a counter/timer, or A/D excursions), you must exercise caution in halting the host processor, disabling Q-Bus interrupts, or entering the Q-Monitor's keyboard command processors. Any of these actions will interfere with proper interrupt generation and servicing.

2.2.2.4 Reboot - The Q-Monitor toggles BDCOK, causing a reboot of the host system.

2.2.2.5 Alarm - The Q-Monitor asserts the Alarm line.

2.2.2.6 Transmit Message - This feature is not yet implemented.

2.2.2.7 Failover - The Q-Monitor asserts the Failover Output line, notifying a redundant host system of a failure. Failover Out can be either active high or active low (the default).

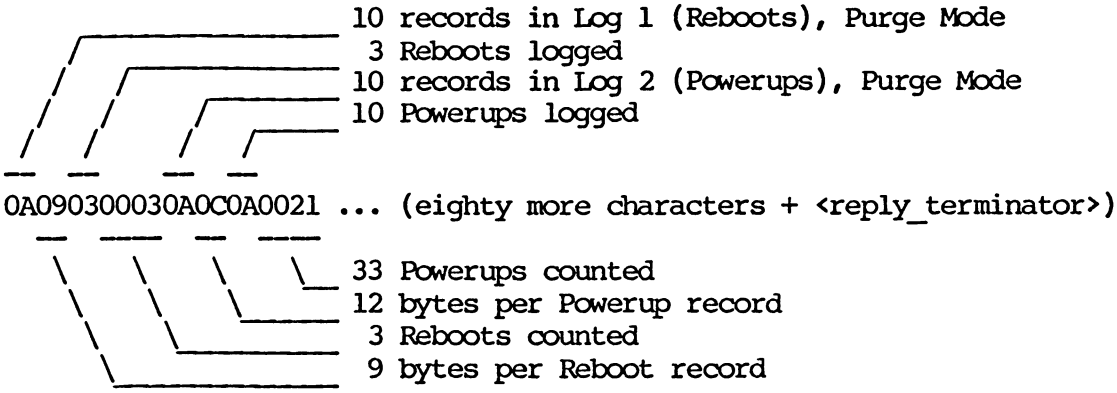
2.3 Examining Logs

The SHOW TABLE and SHOW RECORD commands can be used to examine the Q-Monitor's Event Logs: SHOW TABLE provides information on the record length and the number of records filled in each Log; it also returns the contents of the Event counters. SHOW RECORD can then be used to read individual records from any Log.

2.3.1 SHOW TABLE

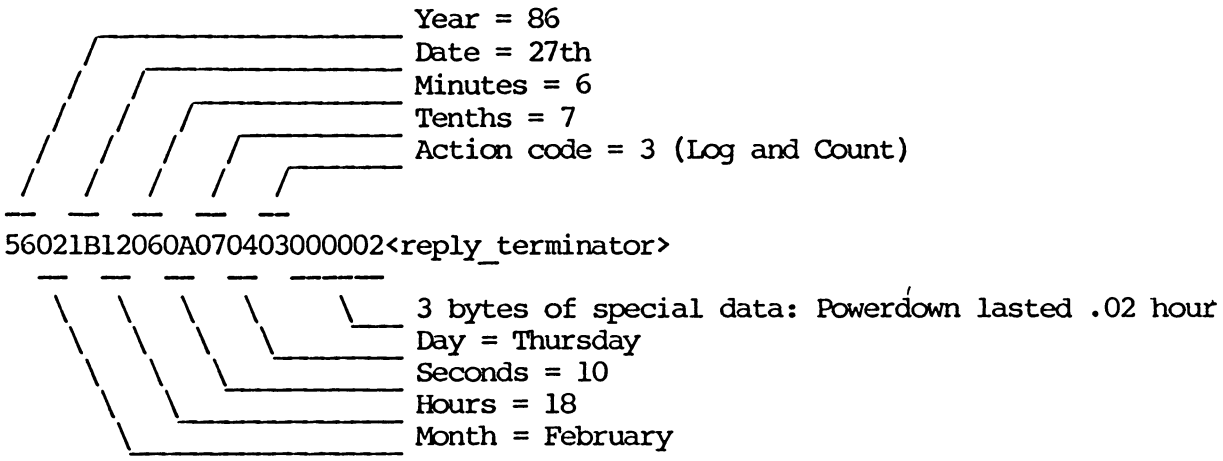
SHOW TABLE returns ten hexadecimal characters for each Event Log, specifying: the number of records in the Log; the number of bytes in each record; the number of records that have been filled; the number of times the Event has oc-

curred; and the type of logging (Purge mode or Stop mode). For example:


10 records in Log 1 (Reboots), Purge Mode
3 Reboots logged
10 records in Log 2 (Powerups), Purge Mode
10 Powerups logged
0A090300030A0C0A0021 ... (eighty more characters + <reply_terminator>)
33 Powerups counted
12 bytes per Powerup record
3 Reboots counted
9 bytes per Reboot record

2.3.2 SHOW RECORD

SHOW RECORD returns a string of hexadecimal characters (the length differs as a function of which Log the record is from) which represent: the date and time the Event occurred; the Action code in effect at that time; and any special data pertinent to this type of Event. For example, if you read a record from Log 2 (Powerups), the data might look like this:



which indicates that the system powered up on THU 27-FEB-86 18:06:10.7 after being down .02 hours, and that the Action taken was just to count and log the powerup.

Special data recorded for the various Events are as follows:

Event	Special Data	Event	Special Data
Reboots	—	A/D excursion	2 bytes: chan'l number, chan'l data value
Powerups	3 bytes: powerdown duration (.01 hour units)	Digital I/O	1 byte: Port data value
W'dog A	—	CT1	—
W'dog B	—	CT2	—
UPS	—	Failover	—

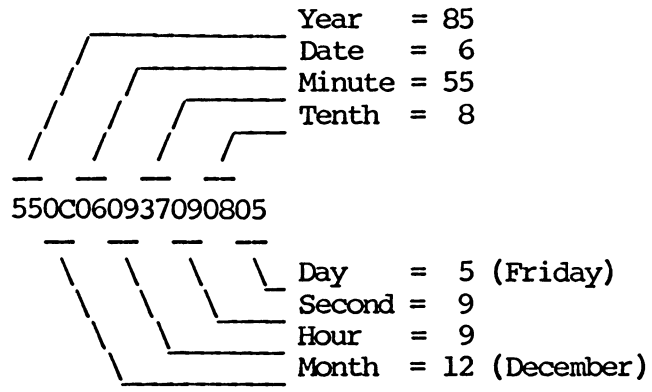
CALENDAR CLOCK

CHAPTER 3

Date and time information from the Q-Monitor's calendar clock is accessible via the SET and SHOW CLOCK escape sequences. These commands communicate the date and time via strings of 16 ASCII hexadecimal characters which represent 8 bytes of binary data. The character positions in these strings, and the values they represent are as follows:

Character	Byte	Value	Character	Byte	Value
0, 1	0	Year	8, 9	4	Minute
2, 3	1	Month (1=January, 12=December)	10, 11	5	Second
4, 5	2	Date	12, 13	6	Tenth
6, 7	3	Hour	14, 15	7	Day (1=Monday, 7=Sunday)

Even numbered characters (0, 2, 4, ... ,14) correspond to the high nibbles of the bytes, while odd numbered characters are the low nibbles. For example, the string "550C060937090805" represents the 8 decimal values 85, 12, 6, 9, 55, 9, 8, and 5 as follows:



or, 1985 December 6th, 9:55:09.8, Friday.

The 16-character string is used for communication with the host processor (e.g. SET CLOCK and SHOW CLOCK), while the 8-byte array is used internally by the Q-Monitor. For readability, QMsetup uses an extension of the Digital convention, and formats this as

FRI 6-DEC-85 9:55:09.8

WATCHDOG TIMERS

CHAPTER 4

The Q-Monitor's watchdog timer is a 16-bit down counter which is decremented at a 10 Hz rate. If the watchdog reaches zero without being reset by a program in the host system, any enabled Action will be taken.

The watchdog timer has two stages: Watchdog A is active from the moment the system is powered up (or the host system's Restart switch is released) until its count reaches zero or until it is reset; Watchdog B is active from the termination of Watchdog A until the system is restarted.

Each constant (one for Watchdog A, one for Watchdog B) can be set to any 16-bit number from 0 to 65535 using the SET WATCHDOGS command. Setting a constant to 0 has the effect of setting it to 65536; hence the intervals that can be selected range from .1 sec to 1:49:13.6.

Any Action can be selected for Watchdog B; any Action except "Interrupt" can be set for Watchdog A. Since you can specify independent constants and Action codes for Watchdogs A and B, you can use them to produce quite specific results depending upon whether the system fails during normal operation, or during a powerup or restart attempt.

Resetting the active Watchdog timer is accomplished either by issuing the RESET WATCHDOG command, or by reading the transmit data buffer (TBUF).

NOTE

- SET WATCHDOGS changes the constants that are used to load the watchdog counter; it does not change the value that is currently in the watchdog counter.
- Selecting an interval shorter than a few seconds is not recommended, unless you are intimately familiar with the behavior of your host system.

BLANK

COUNTER/TIMERS

CHAPTER 5

These two devices (CT1 and CT2) can be used as simple timers and waveform generators by driving them with the internal 250 KHz (4 microsec) clock, or as counters or timers accessed either externally via the digital I/O port or "internally" via the digital I/O port (to count "software" events). For each counter/timer, you can use

SET TIMER CONSTANT	to set its time constant
SET TIMER MODE	to select its duty cycle and to enable retriggering or external access
SET TIMER GCB	to gate the counter/timer under software control
SET TIMER TCB	to trigger or retrigger it under software control
SHOW TIMER COUNT	to read its current count register
SHOW TIMER CSR	to examine the status of the "Gate Command" and "Count in Progress" bits

They can be used independently, or in one of three linked modes (see SET TIMER LINK):

Mode	
0	CT1 and CT2 are independent
1	CT1's output gates CT2
2	CT1's output triggers CT2
3	CT1's output is CT2's input

External access to CT1 and CT2 is available via the digital I/O port as follows:

	Port Bit		Port Bit
CT1 count output	4	CT2 count output	0
CT1 count input	5	CT2 count input	1
CT1 trigger input	6	CT2 trigger input	2
CT1 gate input	7	CT2 gate input	3

NOTE

- When using CT1 and CT2 in the linked modes, it is important to initialize and enable them in the proper sequence
- When configured in any of the linked modes, CT1 cannot generate any Action
- If external access is to be used, you must first configure the digital I/O port appropriately (see SET PORT CONFIGURATION and SET PORT BUFFER)
- It is beyond the scope of this manual to describe all of the possible configurations and uses of CT1 and CT2
- See Part I, Appendix F for further information on digital port configuration

Counter/Timer 1 (CT1) can be used to drive the Q-Bus BEVNT line. There is currently no explicit support for this capability; you must use the SET TIMER commands to program CT1 and turn it on when your system powers up, and you must use the SET PORT commands to configure the digital I/O port to gate CT1's signal to the BEVNT line.

For example, to generate a 50 Hz clock:

```
SET PORT BUFFER 1 or 2          digital port bit <4> must be an output
SET PORT CONFIGURATION ...

SET TIMER LINK 0                make sure CT1 and CT2 are independent,
SET TIMER 1 MODE C2 (16)        select continuous square wave,
                                enable count output,
SET TIMER 1 CONSTANT 2500 (10)  set constant for period of 10 msec
                                (square waves's period is twice that
                                of the counter/timer),
SET TIMER 1 TCB                 and start it running
```

NOTE

Future Q-Monitor PROM releases may include specific BEVNT clock support.

A/D CONVERTER

CHAPTER 6

The Q-Monitor's A/D converter is a relatively low speed 8-bit 8-channel device. The sampling scheme is straightforward: every 10 msec the Q-Monitor samples one of the eight channels in a "round-robin" fashion; thus each channel is sampled, and each instantaneous data value is updated, once every 80 msec. No averaging or scaling is performed and each channel is always sampled, whether or not it is connected to anything.

Every 100 msec the most recent values for the 8 channels are scanned, the maximum and minimum for each channel are revised, and if an excursion is detected (i.e. if the current value falls outside the thresholds) on any channel the specified Action(s) is taken. If an excursion is detected on more than one channel, the Action(s), including the generation of Q-Bus interrupts, will be taken once for each such channel. If an excursion persists, the specified Actions(s) will be repeated every 100 msec until conditions change.

If the Action code specifies that host interrupts be generated, RBUF<2:0> will contain the channel number on which the excursion exists; RBUF<7:3> are reserved and should be masked off.

Channel assignments and conversion factors are:

Channel	Function	Function if Failover is enabled
0	user, 0-5.35 V 21 mV/LSB	redundant system's +12V 63 mV/LSB
1	user, 0-5.35 V 21 mV/LSB	redundant system's +5V 21 mV/LSB
2	user, 0-5.35 V 21 mV/LSB	
3	user, 0-5.35 V 21 mV/LSB	redundant system's temperature .292 degree C/LSB
4	user, 0-5.35 V 21 mV/LSB	
5	host system's +12V supply 63 mV/LSB	
6	host system's +5V supply 21 mV/LSB	
7	host system's temperature .292 degree C/LSB	

A/D converter parameters (thresholds and conversion factors) can be established via SET CONVERTER PARAMETERS; data values are reinitialized by RESET CONVERTER DATA.

A/D converter data (instantaneous values, maxima, and minima) are sampled with SHOW CONVERTER DATA; parameters can be examined via SHOW CONVERTER PARAMETERS.

NOTE

Conversion factors must be included in SET CONVERTER PARAMETERS and are returned by SHOW CONVERTER PARAMETERS, but are not currently implemented.

DIGITAL I/O

CHAPTER 7

The Q-Monitor's digital I/O port can be used to drive external outputs and to monitor external inputs. It is also used to provide external access to the Q-Monitor's counter/timers. Although internally this port has eight lines which can be programmed in any combination of inputs and outputs, some restrictions are imposed on its use by programs in the host system:

- Line <3> of this port is used by the Q-Monitor, is not present on any external connector, and is not altered by any SET PORT commands.
- Because of the limitations of available buffer chips, the port can only be configured as eight inputs (SET PORT BUFFER 0), four inputs/four outputs, (SET PORT BUFFER 1) or eight outputs (SET PORT BUFFER 2).
- For bi-directional buffer chips, SET PORT BUFFER 0 or 2 drives the DIR line appropriately (see Part I, Appendix D)
- If UPS monitoring is enabled, BUFFER=2 is invalid, and one of the input lines (<5> by default) is used for the UPS input
- If Failover Output is enabled, BUFFER=0 is invalid, and one of the output lines (<2> by default) is used for Failover Out

7.1 Port Data

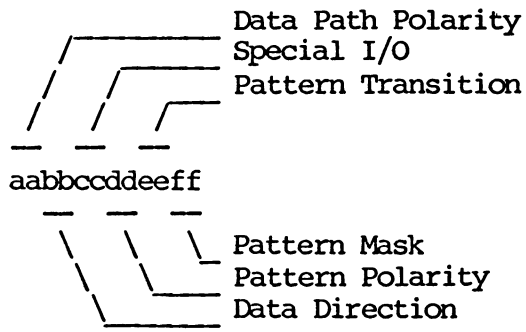
Digital port data can be examined with SHOW PORT DATA, and can be set with SET PORT DATA, subject to the restrictions outlined above.

Reading the port shows the current condition of each line (prior to any inversion), including any reserved lines and any that are programmed as outputs.

7.2 Port Configuration

Digital I/O is controlled by six 8-bit registers: Data Direction, Data Path Polarity, Special I/O, Pattern Polarity, Pattern Transition, and Pattern Mask. In each of these registers, bits <7:0> correspond to port lines <7:0>. These six bytes are represented by the twelve hex characters in the SET/SHOW PORT CONFIGURATION commands, in the order listed above. Specifically,

<ESC> ! I a a b b c c d d e e f f <DEL> (SET PORT CONFIGURATION)



Any input line can be configured with a "1's catcher", which latches "1" conditions which occur on that line.

Each line can be programmed as inverting or non-inverting, and can be made to generate an Action on detection of any transition or level.

7.2.1 Data Direction

Data Direction defines the direction (1 for input, 0 for output) of each of the digital I/O lines. This register is set automatically by the SET PORT BUFFER command, and is not changed by SET PORT CONFIGURATION if BUFFER = 0, 1 or 2.

7.2.2 Path Polarity

Path Polarity defines the polarity (1 for inverting, 0 for non-inverting) of each of the digital I/O lines. This register is set by SET PORT CONFIGURATION.

7.2.3 Special I/O

Special I/O controls "1's catchers" (1 for enable) for each of the digital input lines. This register is set by SET PORT CONFIGURATION, and has no effect for output lines.

7.2.4 Pattern Matching (Pattern Polarity, Transition and Mask registers)

Any digital I/O line (input or output) can be programmed to detect any level (high or low) or transition (low-to-high, high-to-low, or either). This is controlled by the contents of the three pattern match registers, which are the Pattern Polarity register, the Pattern Transition register, and the Pattern Mask register. Pattern sensitivities are enabled as follows:

Pattern Mask	Pattern Transition	Pattern Polarity	Pattern Specification
0	0	X	Bit masked off
0	1	X	Any transition
1	0	0	Zero (level)
1	0	1	One (level)
1	1	0	One-to-Zero transition
1	1	1	Zero-to-One transition

7.3 Buffer Type

The SET PORT BUFFER command selects internal Q-Monitor software support for one of the digital I/O buffer chip options as follows:

SET PORT BUFFER 0 <7:4,2:0> are inputs, DIR/$\overline{EN}$ = 0
SET PORT BUFFER 1 <7:5> are inputs, <4,2:0> outputs, DIR/$\overline{EN}$ = 0
SET PORT BUFFER 2 <7:4,2:0> are outputs, DIR/$\overline{EN}$ = 1

The SET PORT BUFFER parameter must match the type of buffer chip installed; for details, including the assignment of port bits to pins on connector JB, see Part I, Appendix D.

These are the only options currently supported; other buffer types may be available in future PROM releases.

7.4 Port Protection

SET PORT PROTECTION lets you specify lines in the digital I/O port that become exempt from modification by SET PORT DATA and SET PORT CONFIGURATION; it is also used to redefine the I/O lines used to implement UPS monitoring and Failover Output, should this become necessary.

Three bytes of data are required:

SET PORT PROTECTION b1 b2 b3

b1 The port protection byte: no data line corresponding to a "1" in the port protection byte can be written with SET PORT DATA, nor can any of its configuration bits be altered with SET PORT CONFIGURATION.
b2 The UPS selector: the bit which is set indicates the line to be used for UPS Input, when that feature is enabled (no more than one bit may be set).
b3 The Failover Out selector: the bit which is set (if any) indicates the line to be used for Failover Output, when that feature is enabled (no more than one bit may be set).

NOTE

Bit <3> is permanently protected.

HOBBS METER

CHAPTER 8

The Q-Monitor commands SHOW UPTIME and SHOW DOWNTIME can be used to examine these two counters. Each 24-bit counter contains a cumulative time, in .01 hour units, since the last SET BATTERY command was issued (i.e. the SET BATTERY command zeroes both counters)

The uptime Hobbs meter is incremented by the Q-Monitor every 36 seconds when backplane power is on; every time the Q-Monitor is restarted the downtime meter is updated by subtracting the last date and time recorded in the non-volatile RAM from the current time from the calendar clock. Downtimes in excess of 4 years (a leap year cycle) will produce undefined results.

While these "meters" are not perfect, in normal service they will give a good indication of your system's duty cycle and of the Q-Monitor's battery usage.

BLANK

NON-VOLATILE RAM

CHAPTER 9

The Q-Monitor's RAM is not yet available to host programs.

BLANK

TERMINAL EMULATION

CHAPTER 10

To a Q-Bus host processor, the Q-Monitor looks like a DLV11 serial I/O port. Using the terminal driver of your operating system you can communicate with the Q-Monitor itself, or with the terminal connected to it, just as you would with any other terminal port.

I/O with the Q-Monitor's terminal differs in no way from I/O with a terminal connected to a standard serial port.

Output to the Q-Monitor itself (commands) consists of escape sequences; input from the Q-Monitor (data and status) looks just like any other keyboard input, except that it might be much faster if the LIMIT feature is disabled.

10.1 Registers and Vectors

Communication with, and control of, the Q-Monitor are accomplished via the four serial I/O registers

RCSR Receiver Control and Status Register
RBUF Receiver Data Buffer
TCSR Transmitter Control and Status Register
TBUF Transmitter Data Buffer and Watchdog Reset Register

Bit assignments for the registers are generally as for a DLV11-J port (see Part I, Appendix C).

In the standard configuration RCSR is at 176530, which will conflict neither with a normally configured DLV11-J nor with a second DLV11-J at 176540. The Q-Monitor's addressing can be changed by swapping the address PAL located in U14.

By default, the Q-Monitor uses host memory locations (330,332) for the receive interrupt vector and (334,336) for the transmit vector. These can be redefined by issuing the SET VECTORS command from QMsetup or from a host program, subject to the following constraints:

o the receive vector must be a multiple of 8,
o the receive vector must be between 0 and 370 octal, and
o the transmit vector is always at the receive vector + 4

By changing the vector definitions and the address PAL, the Q-Monitor can be used as the console terminal (RCSR = 177560, receive vector = 60).

10.2 Baud Rates

When the Q-Monitor communicates with its terminal, it does so at the speed selected with the SET BAUD command (QMsetup or host); when it communicates with the host processor, it does so at the speed of the "handshake" logic in the host's terminal driver (interrupts or programmed I/O). Because the terminal drivers of many operating systems cannot handle high speed input, the Q-Monitor has two features (LIMIT and STUTTER) to limit the rate at which it inputs data to the host.

Input from the keyboard, or data and status strings generated by the Q-Monitor itself, are placed in a 256 byte ring buffer and then sent to the host at a speed which depends upon three factors:

o the state of the input limit feature (see SET LIMIT), and
o the speed of the terminal connected to the Q-Monitor (see SET BAUD), and
o whether or not the characters are being echoed by the host

LIMIT = 0:

If LIMIT is disabled and the host is echoing the data, then input is at the speed of the terminal; if the host is not echoing the data, then input is at the rate at which the host removes data bytes from the receiver data buffer.

LIMIT = 1:

If LIMIT is enabled and the host is echoing the data, then input is at the speed of the terminal up to a maximum of 100 char/sec; if the host is not echoing the data, then input as at 100 char/sec.

The STUTTER feature is designed to avoid the problem which arises when a host program is unable to post a read promptly after the write which transmits the Q-Monitor command.

When STUTTER is disabled, the Q-Monitor will send the first character of its reply (data or status) to the host at the next "tick" of its 10 millisecond timer, which may occur immediately (within a few microseconds). However, when STUTTER is enabled, it will hesitate at least 10 msec but not more than 20 msec before sending the first character of the reply. In either case, transmission of subsequent characters is unaffected.

10.3 I/O Driver Considerations

Following are some restrictions that apply to programs which communicate with the Q-Monitor. If you are writing your own driver code, you should be familiar with these restrictions; if you are using the terminal driver of one of Digital's operating systems you may ignore them:

- You may not used polled input and interrupt output concurrently
- When "Receiver Done" is asserted, you must remove the character from RBUF before starting any output to the Q-Monitor

10.3.1 RT11

If you are running RT11, two easy ways to install your Q-Monitor are either as "TT:", or as "LS:" (using RT's SET VECTOR and SET CSR commands). The Q-Monitor can be used as "TT:" whether or not it is configured as the console terminal.

The RT/Support Kit includes a program called "SWITCH" which can be used to switch RT11's "TT:" back and forth between the actual console terminal and the Q-Monitor, when the Q-Monitor is configured at its standard addresses and vectors.

Other techniques, such as writing special drivers or using RT's multi-terminal feature, are beyond the scope of this manual.

10.3.2 RSX Variants

If you are running an RSX variant, you can substitute the Q-Monitor for an existing terminal interface (including the console terminal), or alter the CSR address and vector RSX expects to use for another terminal port. Although the details of this latter procedure are beyond the scope of this manual, it involves examining RSX11M.MAP to determine the address (the symbol will be "$TTnn") of terminal TTnn's Status Control Block (SCB), then using ZAP to modify the CSR and vector offsets within the SCB in RSX11M.SYS. Again, we can provide more information on this procedure should you find it necessary after consulting the "Guide to Writing an I/O Driver" and the "Utilities" manuals.

If neither of these methods is appropriate, you can perform a SYSGEN to add a "slot" for the Q-Monitor in the terminal driver. However, RSX offers at least four other techniques for accessing the Q-Monitor, should you choose not to use the terminal driver:

1) A dedicated device driver
2) A task which uses the Connect-to-Interrupt-Vector directive (CINT$)
3) A task which links to a device partition (cannot use interrupts)
4) A task with privileged mapping (cannot use interrupts)

Each of these techniques offers a different combination of capabilities, ease of

use, and ease of implementation.

The RSX/Support Kit includes an example program which uses the CINT$ directive, and a version of the SETUP program which uses a device partition to access the Q-Monitor.

10.3.3 MicroVMS

Digital has chosen not to support DLV11 serial interfaces under the MicroVMS terminal driver. However, the MicroVMS/Support Kit provides access to the Q-Monitor via physical page mapping.

10.3.4 ELAN

ELAN supports DLV11 interfaces, and there are no known restrictions on the use of the Q-Monitor.

KEYBOARD COMMANDS

CHAPTER 11

If the Q-Monitor's BREAK recognition is enabled (see SET BREAK), typing BREAK suspends normal host/Q-Monitor communication, and prompts with

QM_

You may respond with

S to enter QMsetup (to change Q-Monitor parameters)
D to enter QMdebug (a maintenance tool), or
P to resume normal terminal emulation

For example,

```
<BREAK>              <BREAK>              <BREAK>
QM_ S                QM_ D                QM_ P
QMsetup_             QMdebug_              .
 .                    .                    .
 .                    .                    .
 .                    .
```

"QM_" also provides commands for controlling your host system:

CTRL B reboots the host system (momentarily asserts BDCOK)
CTRL H halts the host (asserts BHALT)
CTRL P performs the ODT "P" command, after an implicit ^R
CTRL R releases BHALT ("unhalts" the host)

After each of these three commands the Q-Monitor resumes terminal emulation, allowing the use of the Q-Monitor with "Console ODT" (if it is the console terminal).

NOTE

o CTRL B, CTRL H and CTRL R are effective even if the Q-Monitor is not the console terminal interface; CTRL P should not be used unless the Q-Monitor is the console terminal.
o The current version of the Q-Monitor PROM restricts the use of ODT as follows: if console keyboard interrupts are enabled while the host is halted the BHALT line must remain asserted; i.e. you must neither re-

turn the Halt switch to the run position, nor execute the Q-Monitor's CTRL R command, prior to issuing the CTRL P command. If keyboard interrupts are disabled while the processor is halted, the BHALT line may be released at any time.

o While the Q-Monitor is processing keyboard commands, normal host/Q-Monitor communications and terminal emulation are suspended.
o When the Q-Monitor is configured to generate repetitive Event interrupts (e.g. a counter/timer, or A/D excursions), you must exercise caution in halting the host processor, disabling Q-Bus interrupts, or entering the Q-Monitor's keyboard command processors. Any of these actions will interfere with proper interrupt generation and servicing.
o SET BREAK can be used to disable BREAK recognition, to prevent unauthorized access to these features.

11.1 QMsetup

QMsetup lets you alter Q-Monitor configuration parameters (in either the Event Table or Parameter Page) from the keyboard, without need of a program running in your host system. Currently only a minimal subset of Q-Monitor commands is available in QMsetup; future releases of the Q-Monitor PROM will include an expanded QMsetup.

QMsetup commands are detailed in Part I.

11.2 QMdebug

The Q-Monitor debugger is a development and maintenance tool; no claim as to its correctness or reliability is asserted; any use of it is at your own risk.

QMdebug commands are:

B enters BASIC/DEBUG (if included in ROM); after this command, normal Q-Monitor execution can only be restored by a hardware restart, or by a BASIC statement equivalent to "GOTO @12".

H conditions the terminal emulator to look for a <NUL> as the start cue for a Hex download. Any number of characters can be typed between the "H" and the <NUL>. After the <NUL>, one more character must be typed (typically the <CR> at the end of a host system command line), and then incoming data will be treated as Intel Hex records and loaded into Q-Monitor RAM, while being echoed on the terminal. If a non-zero transfer address is specified in the End Record, the Q-Monitor will jump to that address; otherwise, it resumes terminal emulation.

P continues execution of the interrupted program (normally the terminal emulator)

Rx\ opens one of the interrupted programs's working registers (0 - F)

T restarts the terminal emulator

$xx\ opens one of the interrupted program's control registers:

$IM	Interrupt Mask
$IP	Interrupt Priority (write only)
$IR	Interrupt Request
$F	Flags
$PH	PC High
$PL	PC Low
$R	Register pointer
$SH	Stack pointer High
$SL	Stack pointer Low

%xx\ opens a general register (0 - FF)

xxxx\ opens a memory location (0 - FFFF)

xxxxG starts a program at xxxx

NOTE

- o <CR> closes the current location; <LF> closes it and opens the next consecutive location
- o Although QMdebug will accept input values larger than 8 bits, it presently does not implement word operations except for memory addresses
- o Modifying, or even just examining, certain general registers or device registers may cause unpredictable Q-Monitor behavior

BLANK

THEORY OF OPERATION

CHAPTER 12

The Q-Monitor consists of a very specialized event driven real-time program executing on a Z8 microprocessor. All Z8 instructions (the Q-Monitor's monitor) are contained in a single EPROM; all data (both temporary and permanent) are stored in a CMOS RAM. While all Q-Monitor RAM is provided with battery backup, portions of it are used simply as scratch memory.

On powerup, the Q-Monitor performs its self-initialization, delays for a short user specified duration (see SET DELAY), prints an optional startup message (see SET GREETING), and begins terminal emulation.

NOTE

- o Loss of information from RAM is very unusual as long as the board remains in the backplane; if it is subjected to frequent removal and handling, occasional disruptions can occur.
- o If the Event Table is corrupt, the Q-Monitor reloads the factory values and prints "QMinit -- Event Table reloaded"
- o If the non-volatile Parameter Page is corrupt, the Q-Monitor loads the factory values into the operational parameter page and prints "QMinit -- Parameter Page corrupt"
- o If the non-volatile RAM storing the last date and time, Hobbs meters, startup message, etc. is corrupt, the Q-Monitor prints "QMinit -- NVRAM corrupt"
- o In all cases, the startup message (GREETING) is overwritten by the error message and will have to be redefined using the SET GREETING command

12.1 Terminal Emulation and Communications

The Q-Monitor's terminal emulation is its central design feature: all of its functions are controlled by communicating with the Q-Monitor as if it were a terminal and, of course, communication with the terminal attached to the Q-Monitor is via this mechanism. Terminal emulation is active at all times except during execution of the on-board command processors (entered via BREAK); while any of these is active, no host/Q-Monitor interaction can occur.

Communication with the Q-Monitor is generally as follows: a program in the host system sends a Q-Monitor command (an escape sequence), then reads the Q-Monitor's reply (a data or status string). Input from the Q-Monitor is just like any other input from the keyboard, except that it only occurs in response to a command from the host; there is never any "unsolicited" input from the Q-Monitor.

12.2 Event Interrupts

The Q-Monitor is, however, capable of generating interrupts other than those associated with the terminal. These are the Event interrupts: A/D converter excursions, digital I/O pattern matches, Watchdog timeouts, etc. The Event interrupts are independently programmable: corresponding to each of them is a unique interrupt enable bit, Q-Bus vector, and Q-Bus priority.

An Event interrupt signifies that an incidence of the corresponding Event type has just occurred. If it is a digital I/O interrupt, the contents of the 8-bit port data register are placed in RBUF and must be read by the interrupt service routine; if it is an A/D interrupt, the number of the channel experiencing the excursion is placed in RBUF<2:0> and must be read by the interrupt service routine (RBUF<7:3> are reserved and should be masked off); for all other Event types, RBUF is not used and should not be read.

NOTE

- If you have specified the same vector for more than one Event, you can use SHOW INTERRUPT to determine the Event type when an interrupt occurs.
- You should not specify the same interrupt service routine both for Events which return data in RBUF and for those which do not, as RBUF MUST be read if it contains data and MUST NOT be read if it does not.

12.3 Non-Volatile Storage

The Q-Monitor's RAM is used to store several kinds of non-volatile information, as well as normal operating data such as the stack and temporary variables:

12.3.1 Dates and Times

Every 100 msec, the Q-Monitor reads its calendar clock and records the current date and time. This is the date and time that is returned by a SHOW CLOCK, and it is also used on powerup to determine how long the power has been off. Other items in this group are the battery installation date, the uptime and downtime Hobbs meters, and the reference date and time.

12.3.2 Parameter Page

A group of parameters, known collectively as the Parameter Page, controls all Q-Monitor behavior (except Event Logging) such as baud rate, watchdog timer intervals, counter timers, and digital I/O parameters.

If you have used one of Digital's VT100 terminals, you will be familiar with the way the Parameter Page is maintained. At any given time there are actually three complete sets of these parameters:

- the factory default settings which are in ROM and cannot be modified;
- the "permanent" settings which are in a reserved area of RAM and are modified only by the SAVE command; and
- the current settings which can be changed with the SET command.

If you need to alter any of these parameters you must first change the current settings, then use the SAVE command if you wish to make the changes permanent.

After a powerup or hardware restart, the Q-Monitor performs a check to determine if the permanent settings are valid; if so, it loads them as the current settings; if not, it prints

"QMinit -- Parameter Page corrupt"

and loads the factory settings as the current Parameter Page.

NOTE

The corrupt "permanent" Parameter Page is not modified; you must use SAVE to explicitly overwrite it.

12.3.3 Event Table

The Event Table controls the Q-Monitor's Event logging and responses. Unlike the Parameter Page, only two copies are maintained:

- the factory default settings which are in ROM and cannot be modified; and
- the current settings, which can be changed with the SET command.

For the Event Table, there is no distinction between "current" and "permanent" settings; changes to the Event Table parameters made with the SET command take place immediately and remain in effect until you alter them again. As with the Parameter Page, the Event Table is checked for validity at startup, and if it is found to be invalid "QMinit -- Event Table reloaded" is printed, and the factory settings are loaded.

NOTE

The corrupt Event Table is overwritten with the factory default settings; subsequent restarts will NOT generate the "QMinit -- Event Table reloaded" message, but you will have to redefine any non-standard settings that were previously in effect.

12.3.4 Event Logs

There are ten Event Logs, one for each Event type, each of which consists of 0 or more records containing the date, time, Action code, and other information pertinent to an occurrence of an Event. The sizes of the Event Logs are collectively and simultaneously defined with the SET TABLE command.

12.3.5 User Data

Portions of the RAM which are not used by the Q-Monitor for its own operation (including the data structures described above) are not yet available for storing user data via the STORE and FETCH commands.

PROGRAMMING THE Q-MONITOR

CHAPTER 13

NOTE

In this manual you will see the terms <initiator>, <reply_terminator> and <terminator>; the meanings of these are always as follows:

- o <initiator> is the first character after <ESC> in a host-to-Q-Monitor escape sequence
- o <terminator> is the last character in a host-to-Q-Monitor escape sequence
- o <reply_terminator> is the last character in a Q-Monitor-to-host data or status string

Communication with the Q-Monitor itself (as distinct from communication with the terminal attached to the Q-Monitor) is accomplished via "commands": the host sends a command to the Q-Monitor, waits for the Q-Monitor to reply with data or status, and then continues with further processing.

NOTE

Although the host may use any suitable method (polling, event flags, completion routines, ASTs, etc.) for synchronizing with the Q-Monitor, it must accept the Q-Monitor's reply to its last command before sending another.

Commands from the host to the Q-Monitor are always in the form of escape sequences; information returned by the Q-Monitor (either data or status) is always in the form of a string of printable ASCII characters (not escape sequences) terminated by the <reply_terminator> character. <reply_terminator> is the carriage return unless you redefine it with SET TERMINATOR.

The names given to the Q-Monitor commands (e.g. SHOW CLOCK and SET TERMINATOR) are mnemonics used for convenience and for compatibility with QMsetup. Communication between the host system and the Q-Monitor is, in reality, a stream of ASCII escape sequences, data and status bytes.

Host commands to the Q-Monitor look like:

<ESC> <initiator> <id> ... <terminator> where

<ESC>	is the escape character (decimal 27),
<initiator>	is the ASCII code which identifies this as a Q-Monitor escape sequence (usually "!", decimal 33),
<id>	is the ASCII code which identifies the specific Q-Monitor command to be executed,
...	denotes the command's data or parameters (if any), and
<terminator>	is the ASCII code which signals the end of the escape sequence (usually <DEL>, decimal 127)

Q-Monitor command processing is initiated when the 2nd character of an escape sequence is <initiator>. All subsequent characters are considered part of the Q-Monitor command sequence up to and including <terminator>. Command execution does not begin until <terminator> arrives. No character, from <ESC> through <terminator>, is passed to the terminal.

Any escape sequence which does not have <initiator> as its 2nd character is passed to the terminal without modification. The Q-Monitor's escape sequences have been chosen to avoid conflict with most terminals, including the VT100 and VT200 series. Should a conflict be discovered with the escape sequences used by other terminals, the Q-Monitor's escape sequence delimiters can be changed using the SET DELIMITERS command.

When the host processor transmits a command sequence it must wait for the Q-Monitor to signal completion (by replying either with data or with status) before beginning another command sequence.

Status returned by the Q-Monitor is a single ASCII character followed by <reply_terminator>. Status is returned for any command that does not return data, including any command which is expected to return data but which fails to do so because of an error condition. No status is returned when a command returns data to the host, since the reply itself indicates successful completion.

The status characters which may be returned are:

.	Successful completion
?	Invalid command
C	Configuration conflict
P	Bad parameter

Data returned in response to a Q-Monitor command (except possibly SHOW TERMINATOR and SHOW GREETING) are always printable ASCII characters, and usually are strings of hexadecimal digits (0-F); in a few cases data are letters or decimal digits.

If the host program loses synchronization with the Q-Monitor during an escape sequence (due to a pathological condition in the host), resynchronization can be induced by setting Transmit BREAK (bit 0 of the transmit CSR) for 100 msec, and then clearing it. This will cause the Q-Monitor to discard any partial escape sequence that had been transmitted to it, or to abort any command related data

or status that it was sending to the host.

NOTE

o This DOES NOT cause the Q-Monitor to reset the watchdog timer or any other internal logic.
o This will not solve the problem resulting from the host system missing a character from the Q-Monitor if its terminal driver code is too slow (but see SET LIMIT).

Some Q-Monitor commands require input parameters (situated between <id> and <terminator>) to control subsequent command processing, some do not. Some commands return data to the host; commands which do not return data (including those which abort because of an error condition) return a status byte. Commands which return data do not return a status byte. All Q-Monitor commands return SOMETHING to the host at the end of command execution, to provide the host with a positive handshake mechanism. In all cases, Q-Monitor information (data or status) terminates with <reply_terminator>.

Command codes for which no function is currently implemented return the status character "?", followed by <reply_terminator>.

With the exception of the user startup message (which can include non-ASCII codes 128 through 255 decimal), data transmitted to or from the Q-Monitor are always valid 7-bit ASCII codes.

BLANK

COMMAND DESCRIPTIONS

CHAPTER 14

For each command, a description is given of the data or parameters that must be passed to the Q-Monitor, or that are returned by the Q-Monitor to the host.

For the sake of brevity, these descriptions show "!" as <initiator> and <DEL> as <terminator>; these are the defaults, but others can be selected using SET DELIMITERS.

<lo_item>	denotes the hexadecimal character representing the low nibble of "item"
<hi_item>	denotes the hexadecimal character representing the high nibble of "item"
<ascii_x>	denotes the ASCII character code for "x"
LSB-	denotes the Less Significant Byte of a 16-bit value
MSB-	denotes the More Significant Byte of a 16-bit value

NOTE

- Section headers correspond to the command codes (e.g. 15.1 is <SOH>, 15.32 is <SP>, etc.)
- Unless otherwise noted, data or status returned by a command are terminated by <reply_terminator>
- <NUL> is never used
- <terminator> and <reply_terminator> are two distinct character codes: <reply_terminator> is the last character of Q-Monitor-to-host data or status strings; <terminator> is the trailing delimiter for host-to-Q-Monitor commands

14.1 Reserved

<ESC> ! <SOH> ... <DEL>

Status: ? Invalid command

14.2 Reserved

<ESC> ! <STX> ... <DEL>

Status: ? Invalid command

14.3 Reserved

<ESC> ! <ETX> ... <DEL>

Status: ? Invalid command

14.4 Reserved

<ESC> ! <EOT> ... <DEL>

Status: ? Invalid command

14.5 Reserved

<ESC> ! <ENQ> ... <DEL>

Status: ? Invalid command

14.6 Reserved

<ESC> ! <ACK> ... <DEL>

Status: ? Invalid command

14.7 Reserved

<ESC> ! <BEL> ... <DEL>

Status: ? Invalid command

14.8 Reserved

<ESC> ! <BS> ... <DEL>

Status: ? Invalid command

14.9 Reserved

<ESC> ! <HT> ... <DEL>

Status: ? Invalid command

14.10 Reserved

<ESC> ! <LF> ... <DEL>

Status: ? Invalid command

14.11 Reserved

<ESC> ! <VT> ... <DEL>

Status: ? Invalid command

14.12 Reserved

<ESC> ! <FF> ... <DEL>

Status: ? Invalid command

14.13 Reserved

<ESC> ! <CR> ... <DEL>

Status: ? Invalid command

14.14 Reserved

<ESC> ! <SO> ... <DEL>

Status: ? Invalid command

14.15 Reserved

<ESC> ! <SI> ... <DEL>

Status: ? Invalid command

14.16 Reserved

<ESC> ! <DLE> ... <DEL>

Status: ? Invalid command

14.17 Reserved

<ESC> ! <DC1> ... <DEL>

Status: ? Invalid command

14.18 Reserved

<ESC> ! <DC2> ... <DEL>

Status: ? Invalid command

14.19 Reserved

<ESC> ! <DC3> ... <DEL>

Status: ? Invalid command

14.20 Reserved

<ESC> ! <DC4> ... <DEL>

Status: ? Invalid command

14.21 Reserved

<ESC> ! <NAK> ... <DEL>

Status: ? Invalid command

14.22 Reserved

<ESC> ! <SYN> ... <DEL>

Status: ? Invalid command

14.23 Reserved

<ESC> ! <ETB> ... <DEL>

Status: ? Invalid command

14.24 Reserved

<ESC> ! <CAN> ... <DEL>

Status: ? Invalid command

14.25 Reserved

<ESC> ! <EM> ... <DEL>

Status: ? Invalid command

14.26 Reserved

<ESC> ! <SUB> ... <DEL>

Status: ? Invalid command

14.27 Reserved

<ESC> ! <ESC> ... <DEL>

Status: ? Invalid command

14.28 Reserved

<ESC> ! <FS> ... <DEL>

Status: ? Invalid command

14.29 Reserved

<ESC> ! <GS> ... <DEL>

Status: ? Invalid command

14.30 Reserved

<ESC> ! <RS> ... <DEL>

Status: ? Invalid command

14.31 Reserved

<ESC> ! <US> ... <DEL>

Status: ? Invalid command

14.32 SET CLOCK

Set the Q-Monitor's calendar clock: the escape sequence includes 16 ASCII hex characters representing 8 bytes for year, month, date, hour, minute, second, tenth, and day.

<ESC> ! <SP> <hi_yr> <lo_yr> <hi_mon> <lo_mon> <hi_date> <lo_date>
<hi_hr> <lo_hr> <hi_min> <lo_min> <hi_sec> <lo_sec>
<hi_tenth> <lo_tenth> <hi_day> <lo_day> <DEL>

Status: . Success
P Bad parameter

Notes: o Valid parameters are:

Year	0-99
Month	1-12
Date	1-28, 29, 30, or 31 (as appropriate)
Hour	0-23
Minute	0-59
Second	0-59
Tenth	ignored
Day	1-7 for Monday ... Sunday

14.33 SHOW CLOCK

The Q-Monitor returns 16 ASCII hex characters representing the contents of the calendar clock, as defined in SET CLOCK.

<ESC> ! ! <DEL>

Status: N/A

14.34 SET WATCHDOGS

Set the values of the Q-Monitor's Watchdog A and Watchdog B constants: the escape sequence includes 8 ASCII hex characters, representing a 16-bit number for Watchdog A and one for Watchdog B, respectively.

<ESC> ! " <hi_MSB-A> <lo_MSB-A> <hi_LSB-A> <lo_LSB-A>
<hi_MSB-B> <lo_MSB-B> <hi_LSB-B> <lo_LSB-B> <DEL>

Status: . Success

Notes: o Any two 16-bit numbers are valid
o The current value of the active watchdog's counter is not changed; the new value will not be loaded until the watchdog expires or is reset

14.35 SHOW WATCHDOGS

The Q-Monitor returns 8 ASCII hex characters representing the values of the Watchdog A and Watchdog B constants, as defined in SET WATCHDOGS.

<ESC> ! # <DEL>

Status: N/A

Notes: o The values returned are the respective time constants, and do not reflect the instantaneous value of the active watchdog.

14.36 SET ACTIONS

Set the Action codes for the Q-Monitor's ten Events: the escape sequence includes 20 ASCII hex characters, representing an Action code for each of the Event types.

<ESC> ! $ <hi_reboot> <lo_reboot> <hi_powerup> <lo_powerup>
<hi_W'dog-A> <lo_W'dog-A> <hi_W'dog-B> <lo_W'dog-B>
<hi_UPS> <lo_UPS> <hi_A/D> <lo_A/D> <hi_PIO> <lo_PIO>
<hi_CT1> <lo_CT1> <hi_CT2> <lo_CT2>
<hi_Failover-In> <lo_Failover-In> <DEL>

Status: . Success

Notes: o No validity checking is performed; invalid combinations are loaded, but ignored
o Action code bit assignments are
<7> Reserved
<6> Failover Out
<5> Message
<4> Alarm
<3> Reboot
<2> Interrupt
<1> Log
<0> Count

14.37 SHOW ACTIONS

The Q-Monitor returns 20 ASCII hex characters representing the Actions codes for each of the ten Events, as defined in SET ACTIONS.

<ESC> ! % <DEL>

Status: N/A

14.38 SET REFERENCE

The escape sequence includes 16 ASCII hex characters, representing 8 bytes for year, month, date, hour, minute, second, tenth, day.

<ESC> ! & <hi_yr> <lo_yr> <hi_mon> <lo_mon> <hi_date> <lo_date>
<hi_hr> <lo_hr> <hi_min> <lo_min> <hi_sec> <lo_sec>
<hi_tenth> <lo_tenth> <hi_day> <lo_day> <DEL>

Status: . Success
P Bad parameter

Notes: o All Event log records and counters are cleared
o Valid parameters are:

Year	0-99
Month	1-12
Date	1-28, 29, 30, or 31 (as appropriate)
Hour	0-23
Minute	0-59
Second	0-59
Tenth	ignored
Day	1-7 for Monday ... Sunday

14.39 SHOW REFERENCE

The Q-Monitor returns 16 ASCII hex characters as defined in SET REFERENCE.

<ESC> ! ' <DEL>

Status: N/A

14.40 SET BATTERY

Set the Q-Monitor's battery installation date: the escape sequence includes 8 ASCII hex characters representing the date as four bytes for year, month, date, and day, respectively. This command also zeroes the UPTIME and DOWNTIME Hobbs meters.

<ESC> ! (<hi_yr> <lo_yr> <hi_mon> <lo_mon>
<hi_date> <lo_date> <hi_day> <lo_day> <DEL>

Status: . Success
P Bad parameter

Notes: o Zeroes both Hobbs meters
o Valid parameters are:

Year	0-99
Month	1-12
Date	1-28, 29, 30, or 31 (as appropriate)

Day 1-7 for Monday ... Sunday

14.41 SHOW BATTERY

The Q-Monitor returns 8 ASCII hex characters as defined in SET BATTERY.

<ESC> !) <DEL>

Status: N/A

14.42 SHOW DOWNTIME

The Q-Monitor returns 6 ASCII hex characters representing 3 bytes which constitute a 24-bit number which is the Q-Monitor's downtime in .01 hour units. The first byte is the most significant byte of the 24-bit number.

<ESC> ! * <DEL>

Status: N/A

Notes:
- o The downtime Hobbs meter is a 24-bit counter which is updated by the Q-Monitor each time it performs a cold start. This update consists of subtracting the "old" time in non-volatile RAM from the current time, rounding to the nearest 1/100th hour, and adding the result to the previous value.
- o Downtimes of more than 4 years cause unpredictable results

14.43 RECALL FACTORY

Copy the factory Parameter Page into the current Parameter Page.

<ESC> ! + <DEL>

Status: . Success

14.44 SAVE

Write the current Parameter Page into the "permanent" Parameter Page in non-volatile RAM.

<ESC> ! , <DEL>

Status: . Success

Notes: o This is the only command which causes "permanent" changes to the Q-Monitor Parameter Page; until this command is executed, changes made with SET commands will be lost when the Q-Monitor is restarted

14.45 RECALL SAVED

Copy the "permanent" parameters into the current Parameter Page.

<ESC> ! - <DEL>

Status: . Success

14.46 SET VECTORS

Set the Q-Monitor's interrupt vectors: the escape sequence includes 16 ASCII hex characters representing eight bytes which are the low order portions of the host vector addresses for the SLU receiver, Watchdog B timeouts, UPS status transitions, A/D threshold excursions, digital I/O pattern matches, CT1, CT2 and Failover In transitions.

<ESC> ! . <hi_SLU> <lo_SLU> <hi_W'dog-B> <lo_W'dog-B>
<hi_UPS> <lo_UPS> <hi_A/D> <lo_A/D>
<hi_PIO-vec> <lo_PIO-vec> <hi_CT1> <lo_CT1>
<hi_CT2> <lo_CT2> <hi_Failover-In> <lo_Failover-In> <DEL>

Status: . Success
P Bad parameter

Notes: o Vectors must be in the range 0 to 374 octal, 0 to 252 decimal
o The SLU vector must be a multiple of 8
o The SLU transmit vector is at the SLU receive vector + 4
o Each Event vector must be a multiple of 4
o Event vectors need not be unique
o A value must be specified for each vector, even if not used

14.47 SHOW VECTORS

The Q-Monitor returns 16 ASCII hex characters representing the 8 interrupt vectors, as defined in SET VECTORS.

<ESC> ! / <DEL>

Status: N/A

14.48 SET PRIORITIES

Set the Q-Bus priorities for the Q-Monitor's interrupt sources: the escape sequence includes 16 ASCII hex characters representing 8 bytes which are the interrupt priorities for the SLU transmitter and receiver, Watchdog B timeouts, UPS status transitions, A/D threshold excursions, digital I/O pattern matches, CT1, CT2, and Failover In.

<ESC> ! 0 <hi_SLU> <lo_SLU> <hi_W'dog-B> <lo_W'dog-B>
<hi_UPS> <lo_UPS> <hi_A/D> <lo_A/D>
<hi_PIO-vec> <lo_PIO-vec> <hi_CT1> <lo_CT1>
<hi_CT2> <lo_CT2> <hi_Failover-in> <lo_Failover-in> <DEL>

Status: . Success
P Bad parameter

Notes: o Valid priorities are 4, 5, or 6; priority 7 interrupts are not currently permitted by the Q-Monitor
o SLU transmitter and receiver priorities are always the same

14.49 SHOW PRIORITIES

The Q-Monitor returns 16 ASCII hex characters representing the Q-Bus priorities of its interrupt sources, as defined in SET PRIORITIES.

<ESC> ! 1 <DEL> (the digit "one")

Status: N/A

14.50 SET GREETING

Set the Q-Monitor's optional startup message: the escape sequence includes up to 127 bytes terminated by <DEL> which the Q-Monitor will print on its terminal during power up. The string need not be restricted to 7-bit ASCII codes, but may not contain the codes 127 (<DEL> itself) or 128.

<ESC> ! 2 ... <DEL>

Status: N/A

Notes: o The host system is unaware that this message is printed; the message is printed directly on the local terminal.

14.51 SHOW GREETING

The Q-Monitor returns the startup message: up to 127 characters, terminated by <reply_terminator>.

<ESC> ! 3 <DEL>

Status: N/A

14.52 SET PARITY

Set the Q-Monitor's terminal I/O parity parameter (transmit and receive): the escape sequence includes one ASCII character, either O for Odd parity, or D for parity Disabled.

<ESC> ! 4 <ascii_O-or-D> <DEL>

Status: . Success
P Bad parameter

Notes: o Even parity is not supported
o The Q-Monitor SLU operates only in the 8-data-bit mode
o This parameter affects only Q-Monitor/terminal communications, and has nothing to do with Q-Monitor/host system communications.
o Transmit and receive parity are the same: both odd or both disabled

14.53 SHOW PARITY

The Q-Monitor returns the ASCII letter O or D representing the parity used by its terminal, as defined in SET PARITY.

<ESC> ! 5 <DEL>

Status: N/A

14.54 SET TABLE

Set the Q-Monitor's Event logging parameters: the escape sequence includes 20 ASCII hex characters, representing 10 bytes which constitute the maximum number of records for each of the ten Event types; the value is positive to select Purge mode logging, negative for Stop mode logging, or 0 for no logging.

```
<ESC> ! 6 <hi_nrec-reboot> <lo_nrec-reboot>
          <hi_nrec-powerup> <lo_nrec-powerup>
          <hi_nrec-Watchdog-A> <lo_nrec-Watchdog-A>
          <hi_nrec-Watchdog-B> <lo_nrec-Watchdog-B>
          <hi_nrec-UPS> <lo_nrec-UPS>
          <hi_nrec-A/D> <lo_nrec-A/D>
          <hi_nrec-PIO> <lo_nrec-PIO>
          <hi_nrec-CT1> <lo_nrec-CT1>
          <hi_nrec-CT2> <lo_nrec-CT2>
          <hi_nrec-Failover-In> <lo_nrec-Failover-In>
          <DEL>
```

Status: . Success
P Insufficient memory

Notes: o SET TABLE clears all Event logs
o The length of each Event log must be specified; no partial reconfiguration is allowed

14.55 SHOW TABLE

For each Event, the Q-Monitor returns 10 ASCII hex characters representing 5 bytes, which are the number of records in the Log (2 characters), record length (2 characters), number of records in use (2 characters), and the Event counter (4 characters), in that order. A total of 100 characters, 10 for each of the 10 Events, is returned.

```
<ESC> ! 7 <DEL>
```

Status: N/A

14.56 RESET LOG

The escape sequence includes 2 ASCII hex characters representing one byte which selects the Event log to be cleared.

<ESC> ! 8 <hi_enum> <lo_enum> <DEL>

Status: . Success
P Bad parameter

Notes: o Events codes are:
1 Reboots
2 Powerups
3 Watchdog A timeouts
4 Watchdog B timeouts
5 UPS transitions
6 A/D excursions
7 Digital I/O transitions
8 CT1
9 CT2
10 Failover In

14.57 SHOW RECORD

The Q-Monitor returns a string of ASCII hex characters which represent a record in one of the Event Logs: the escape sequence includes 4 hex characters representing two bytes which specify the record number and the Event number to be returned.

<ESC> ! 9 <hi_record> <lo_record> <hi_event> <lo_event> <DEL>

Status: P Bad parameter

Notes: o Event codes are as defined in RESET LOG
o Valid record numbers are 1 to the highest record number currently in use in the log (this can be determined with SHOW TABLE)
o The number of bytes in a record is always 9 or greater, hence the number of data characters returned is always 18 or greater; the exact length can be determined via SHOW TABLE
o The string always ends with <reply_terminator>
o Special data recorded for the Events are:

Event	Special Data	Event	Special Data
Reboots	—	A/D excursion	2 bytes: chan'l number, chan'l data value
Powerups	3 bytes: powerdown duration (.01 hour units)	Digital I/O	1 byte: Port data value
W'dog A	—	CT1	—
W'dog B	—	CT2	—
UPS	—	Failover	—

14.58 SET MASK

Set the Q-Monitor's Event interrupt enable mask: the escape sequence includes 4 ASCII hex characters representing a two byte bit mask which enables or disables the Event interrupts.

<ESC> ! : <hi_MSB> <lo_MSB> <hi_LSB> <lo_LSB> <DEL>

Status: N/A

Notes: o Watchdog A timeouts cannot cause interrupts
o Bit assignments are:
<15:7> Reserved
<6> Failover In
<5> CT2
<4> CT1
<3> Digital I/O transitions
<2> A/D excursions
<1> UPS transitions
<0> Watchdog B timeouts
o In each case 0 disables the interrupt, 1 enables it

14.59 SHOW MASK

The Q-Monitor returns 4 ASCII hex characters representing the Event interrupt enable mask, as defined in SET MASK.

<ESC> ! ; <DEL>

Status: N/A

Notes: o Bit assignments are as defined in SET MASK

14.60 SET CONVERTER PARAMETERS

Set the parameters which control the Q-Monitor's A/D converter processing: the escape sequence includes 8 ASCII hex characters representing 4 bytes specifying the channel number, its upper threshold, lower threshold and conversion factor.

<ESC> ! < <hi_chan-number> <lo_chan-number>
<hi_upper-t'hold> <lo_upper-t'hold>
<hi_lower-t'hold> <lo_lower-t'hold>
<hi_factor> <lo_factor> <DEL>

Status: . Success

Notes: o No validity checking is performed -- all combinations are valid

14.61 SHOW CONVERTER PARAMETERS

The Q-Monitor returns 48 ASCII hex characters representing 24 bytes as follows: the first 16 bytes are alternating upper and lower thresholds for channels 0-7, in that order; the last 8 bytes are the conversion factors for channels 0-7.

<ESC> ! = <DEL>

Status: N/A

14.62 SHOW CONVERTER DATA

The Q-Monitor returns 64 ASCII hex characters representing 32 bytes as follows: the first 8 bytes are the instantaneous values for channels 0-7, in that order; the next 16 bytes are the alternating maxima and minima for channels 0-7; the last 8 bytes are undefined.

<ESC> ! > <DEL>

Status: N/A

Notes: o All values are unipolar (unsigned) 0 to 255 decimal

14.63 RESET CONVERTER DATA

Reset the instantaneous samples, the maxima and the minima for all eight A/D converter channels to initial values.

<ESC> ! ? <DEL>

Status: . Success

Notes: o Instantaneous values are set to zero
o Maxima are set to 0
o Minima are set to 255

14.64 SET BAUD

Set the baud rate at which the Q-Monitor communicates with its terminal: the escape sequence includes one ASCII digit (1-9) to select the baud rate.

<ESC> ! @ <ascii_1-9> <DEL>

Status: . Success
P Bad parameter

Notes: o Values are:

1	19200 baud		6	600 baud	
2	9600	"	7	300	"
3	4800	"	8	150	"
4	2400	"	9	110	"
5	1200	"			

14.65 SHOW BAUD

The Q-Monitor returns one ASCII digit representing the baud rate as defined in SET BAUD.

<ESC> ! A <DEL>

Status: N/A

Notes: o Baud rates are as shown in SET BAUD

14.66 SHOW UPTIME

The Q-Monitor returns 6 ASCII hex characters representing 3 bytes which constitute a 24-bit number which is the uptime in .01 hour units. The first byte is the most significant byte of the 24-bit number.

<ESC> ! B <DEL>

Status: N/A

Notes: o The UPTIME meter is a 24-bit counter which is incremented by the Q-Monitor every 36 seconds (1/100th of an hour) that it is running.
o Time spent in QM_, QMdebug_ and QMsetup_ is added neither to this counter, nor to the DOWNTIME meter.

14.67 SET LIMIT

Enable or disable the Q-Monitor's 100 character/second host input limit: the escape sequence includes the ASCII digit 0 to disable the limit, or the ASCII digit 1 to enable it.

<ESC> ! C <ascii_0-or-1> <DEL>

Status: . Success
P Bad parameter

14.68 SHOW LIMIT

The Q-Monitor returns the ASCII digit 0 indicating that the 100 char/sec limit is disabled, or the ASCII digit 1 to indicate that it is enabled.

<ESC> ! D <DEL>

Status: N/A

14.69 SET BREAK

Enable or disable the Q-Monitor's BREAK recognition logic: the escape sequence includes the ASCII digit 0 to disable BREAK recognition, or the ASCII digit 1 to enable it.

<ESC> ! E <ascii_0-or-1> <DEL>

Status: . Success
P Bad parameter

Notes: o If BREAK recognition is disable, it can only be reenabled by a program running in the host system

14.70 SHOW BREAK

The Q-Monitor returns the ASCII digit 0 indicating that its BREAK recognition is disabled, or the ASCII digit 1 to indicate that it is enabled.

<ESC> ! F <DEL>

Status: N/A

14.71 SET PORT DATA

Set the condition of the Q-Monitor's digital I/O port lines: the escape sequence includes 2 ASCII hex characters, representing one byte to be loaded into the digital port data register.

<ESC> ! G <hi_data> <lo_data> <DEL>

Status: . Success

Notes: o Bit <3> is always unaffected
o Bits which are in use for UPS input and Failover Out are unaffected

14.72 SHOW PORT DATA

The Q-Monitor returns 2 ASCII hex characters, representing the contents of the digital port data register,

<ESC> ! H <DEL>

Status: N/A

Notes: o Bit <3> is undefined

14.73 SET PORT CONFIGURATION

The escape sequence includes 12 ASCII hex characters, representing the digital port Data Path Polarity, Data Direction, Special I/O, Pattern Polarity, Pattern Transition, and Pattern Mask registers.

<ESC> ! I <hi_DPP> <lo_DPP> <hi_DD> <lo_DD>
<hi_SIO> <lo_SIO> <hi_PP> <lo_PP>
<hi_PT> <lo_PT> <hi_PM> <lo_PM> <DEL>

Status: . Success
C Configuration conflict --
data direction/buffer type mismatch

Notes: o Depending upon the buffer chip installed, the port may be:
8 inputs <7:0>
4 inputs <7:5>,<3> and 4 outputs <4>,<2:0>
1 input <3> and 7 outputs <7:4>,<2:0>
(see SET PORT BUFFER)
o Bit <3> cannot be configured by the user; any specification for this line will be ignored.
o If UPS support is selected Bit <5> cannot be configured by the user, except indirectly via the SET UPS CONFIGURATION command.
o If Failover Out support is selected Bit <2> cannot be configured by the user, except indirectly via SET FAILOVER CONFIGURATION.

14.74 SHOW PORT CONFIGURATION

The Q-Monitor returns 12 ASCII hex characters as defined in SET PORT CONFIGURATION.

<ESC> ! J <DEL>

Status: N/A

Notes: o Bit <3> is undefined

14.75 SET TIMER CONSTANT

Set the time constant for a counter/timer (the value which is loaded into the down counter when the counter/timer is triggered); the escape sequence includes the ASCII digit 1 or 2 to select CT1 or CT2, followed by 4 ASCII hex characters representing the 16-bit time constant for the specified counter/timer.

<ESC> ! K <ascii_1-or-2> <hi_MSB> <lo_MSB> <hi_LSB> <lo_LSB> <DEL>

Status: . Success
P Invalid timer (must be 1 or 2)

Notes: o The time constant is an unsigned 16-bit number; a value of 0 specifies a time constant of 65536
o SET TIMER CONSTANT has no immediate effect on the counter/timer's down counter; the new value will be loaded when the timer next reaches terminal count
o When either counter is driven by the internal 250 KHz clock, one count corresponds to 4 microseconds; for example, for count output modes "pulse" and "one shot":

```
CONSTANT =     1 ...      4 usec
CONSTANT =     2 ...      8 usec
CONSTANT =  2500 ...  10000 usec, or 10 msec
CONSTANT = 25000 ... 100000 usec, or 100 msec
```

o For "square wave" output, the counter must reach terminal count twice for each cycle of the square wave, thus the counts above must be halved to generate a square wave of the periods shown

14.76 SHOW TIMER CONSTANT

The escape sequence includes the ASCII digit 1 or 2 to select CT1 or CT2. The Q-Monitor returns 4 ASCII hex characters representing the 16-bit time constant of the specified counter/timer.

<ESC> ! L <ascii_1-or-2> <DEL>

Status: P Invalid timer (must be 1 or 2)

14.77 SHOW TIMER COUNT

The escape sequence includes the ASCII digit 1 or 2 to select CT1 or CT2. The Q-Monitor returns 4 ASCII hex characters representing the 16-bit current count for the specified counter/timer.

<ESC> ! M <ascii_1-or-2> <DEL>

Status: P Invalid timer (must be 1 or 2)

Notes: o The number returned is the instantaneous value of the down counter (which may be changing) at the moment it was sampled. Using SHOW TIMER COUNT to synchronize with a fast timer is not recommended.

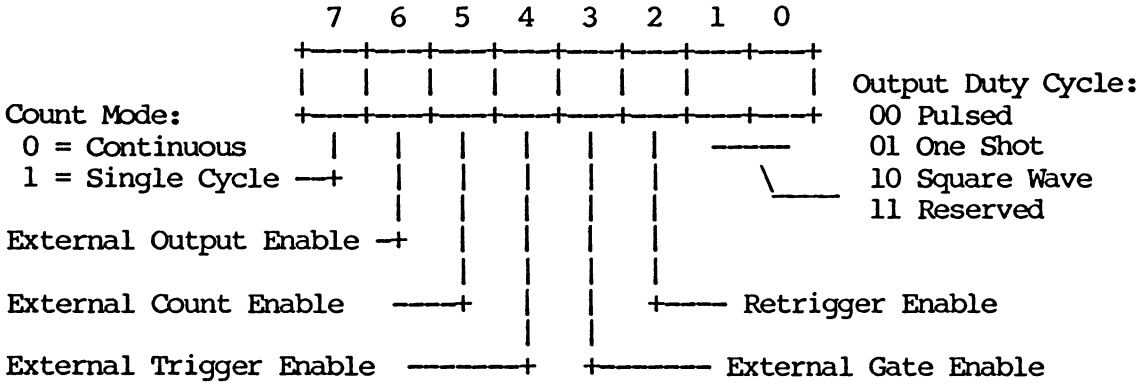
14.78 SET TIMER MODE

The escape sequence includes the ASCII digit 1 or 2 to select CT1 or CT2, followed by 2 ASCII hex characters representing the value to be loaded into the Mode register for the specified counter/timer.

<ESC> ! N <ascii_1-or-2> <hi_mode> <lo_mode> <DEL>

Status: . Success
P Invalid timer (must be 1 or 2)

Notes: o External access to counter/timers requires the digital I/O port
o When square wave output is selected, the counter/timer goes through two full sequences for each cycle of the square wave; thus the wave's period is twice that of the counter/timer
o A counter/timer is retriggerable if and only if its "Retrigger Enable" is set
o Bit assignments in the Mode registers are:



14.79 SHOW TIMER MODE

The escape sequence includes the ASCII digit 1 or 2 to select CT1 or CT2. The Q-Monitor returns 2 ASCII hex characters representing the contents of the Mode register (see SET TIMER MODE) for the specified counter/timer.

<ESC> ! O <ascii_1-or-2> <DEL>

Status: P Invalid timer (must be 1 or 2)

14.80 SET TIMER LINK

The escape sequence includes an ASCII 0, 1, 2, or 3 to set the CT1/CT2 linkage mode.

<ESC> ! P <ascii_0-or-1> <DEL>

Status: . Success
P Bad parameter (must be 0, 1, 2, or 3)

Notes: o CT1 and CT2 can be linked as follows:
0 CT1 and CT2 are independent
1 CT1's OUTPUT (L) gates CT2
2 CT1's OUTPUT (L) triggers CT2
3 CT1's OUTPUT (L) is CT2's count input

14.81 SHOW TIMER LINK

The Q-Monitor returns an ASCII character representing the CT1/CT2 linkage as defined in SET TIMER LINK.

<ESC> ! Q <DEL>

Status: N/A

14.82 SET TIMER GCB

The escape sequence includes the ASCII digit 1 or 2 to select CT1 or CT2, followed by an ASCII 0 to clear or an ASCII 1 to set the Gate Command Bit in the specified counter/timer's CSR.

<ESC> ! R <ascii_1-or-2> <ascii_0-or-1> <DEL>

Status: . Success
P Invalid timer or bad parameter

Notes: o The condition of the Gate Command bit can be determined by examining the value returned by SHOW TIMER CSR
o After powerup, each counter/timer's Gate Command bit is "1"

14.83 SET TIMER TCB

Set the Trigger Command bit of CT1 or CT2; the escape sequence includes an ASCII 1 or 2 to select the counter/timer.

<ESC> ! S <ascii_1-or-2> <DEL>

Status: . Success
P Invalid timer

Notes: o SET TIMER TCB must be specified to start a counter/timer
o SET TIMER TCB will retrigger the counter/timer (i.e. it will reload the TIMER CONSTANT value into the down counter) if and only if "Retrigger Enable" in the Mode register is set (see SET TIMER MODE)
o "Trigger Command" is a write only bit

14.84 SHOW TIMER CSR

The escape sequence includes an ASCII 1 or 2 to select CT1 or CT2; the Q-Monitor returns 2 ASCII hex characters representing the 8-bit Control and Status register of the specified counter/timer.

<ESC> ! T <ascii_1-or-2> <DEL>

Status: P Invalid timer

Notes: o The only CSR bits of interest to the user are:
<0> Count In Progress (set while the counter/timer is counting), and
<2> Gate Command (controlled by SET TIMER GCB)
o After powerup, each counter/timer's Gate Command bit is "1"

14.85 SET UPS CONFIGURATION

The escape sequence includes ASCII H or L to set UPS active high or active low, or ASCII D to disable UPS logging.

<ESC> ! U <ascii_D-H-or-L> <DEL>

Status: . Success
P Bad parameter

14.86 SHOW UPS CONFIGURATION

The Q-Monitor returns ASCII H or L to indicate active UPS input, or ASCII D for UPS logging disabled.

<ESC> ! V <DEL>

Status: N/A

14.87 BOOT

Restarts the host system (toggles Q-Bus BDCOK).

<ESC> ! W <DEL>

Status: N/A

14.88 SET ALARM

The escape sequence includes ASCII 0 to negate, or ASCII 1 to assert the Alarm output.

<ESC> ! X <ascii_0-or-1> <DEL>

Status: . Success
P Bad parameter

14.89 SET FAILOVER OUT

The escape sequence includes ASCII 0 to negate, or ASCII 1 to assert Failover Output.

<ESC> ! Y <ascii_0-or-1> <DEL>

Status: . Success

P Bad parameter

14.90 SET FAILOVER CONFIGURATION

The escape sequence includes ASCII H or L to set Failover Input active high or low; then ASCII H or L to set Failover Input active high or low, or ASCII D to disable disable Failover Output.

<ESC> ! Z <ascii_H-or-L> <ascii_D-H-or-L> <DEL>

Status: . Success
P Bad parameter

14.91 SHOW FAILOVER CONFIGURATION

The Q-Monitor returns ASCII H or L to indicate active Failover Input; then ASCII H or L for active Failover Output, or ASCII D for Failover Output disabled.

<ESC> ! [<DEL>

Status: N/A

14.92 SHOW UPS INPUT

The Q-Monitor returns ASCII 0 or 1 for UPS input False or True.

<ESC> ! \ <DEL>

Status: N/A

14.93 SHOW FAILOVER INPUT

The Q-Monitor returns ASCII 0 or 1 for Failover Input False or True.

<ESC> !] <DEL>

Status: N/A

14.94 SET PORT PROTECTION

The escape sequence includes 6 ASCII hex characters representing 3 bytes to indicate which digital port bits cannot be modified by SET PORT DATA or SET PORT CONFIGURATION.

<ESC> ! ^ <hi_user> <lo_user> <hi_UPS> <lo_UPS>
<hi_Failover-Out> <lo_Failover-Out> <DEL>

Status: . Success
P Bad parameter

14.95 SHOW PORT PROTECTION

The Q-Monitor returns 6 characters representing the 3 port protection bytes as defined in SET PORT PROTECTION.

<ESC> ! _ <DEL>

Status: N/A

14.96 SET PORT BUFFER

The escape sequence includes an ASCII digit which specifies buffer chip support.

<ESC> ! ` <ascii_0-1-or-2> <DEL>

Status: . Success
P Bad parameter

Notes: o Valid parameters are:
0 <7:0> are inputs
1 <7:5,3> are inputs, <4,2:0> are outputs
2 <3> is an input, <7:4,2:0> are outputs
o <3> is always an input, and is never available to the user
o Other buffer options may be made available by special arrangement

14.97 SHOW PORT BUFFER

The Q-Monitor returns one ASCII digit showing the buffer support in effect as defined in SET PORT BUFFER.

<ESC> ! a <DEL>

Status: N/A

14.98 SHOW INTERRUPT

The Q-Monitor returns 2 ASCII hex characters representing the Event code of the Event which caused the last interrupt.

<ESC> ! b <DEL>

Status: N/A

Notes: o This command is useful if you are using the same vector for more than one Event

14.99 SET TERMINATOR

The escape sequence includes 2 ASCII hex characters specifying the 8-bit character code which will terminate all subsequent Q-Monitor data or status inputs.

<ESC> ! c <hi_new-reply-terminator> <lo_new-reply-terminator> <DEL>

Status: . Success

14.100 SHOW TERMINATOR

The Q-Monitor returns a single character, the <reply_terminator>.

<ESC> ! d <DEL>

Status: N/A

14.101 SHOW ID

The Q-Monitor returns 2 ASCII hex characters which specify its PROM version.

<ESC> ! e <DEL>

Status: N/A

14.102 SET DELAY

The escape sequence includes 2 ASCII hex characters which represent an 8-bit number defining the startup delay. This delay is from powerup until the optional startup message is printed, allowing terminals to complete any powerup self tests.

<ESC> ! f <hi_delay> <lo_delay> <DEL>

Status: . Success

Notes: o Delay "units" are approximately .6 second

14.103 SHOW DELAY

The Q-Monitor returns 2 ASCII hex characters which represent an 8-bit startup delay as defined in SET DELAY.

<ESC> ! g <DEL>

Status: N/A

Notes: o Delay "units" are approximately .6 second

14.104 RESET WATCHDOG

Reloads the watchdog down counter with the Watchdog B constant.

<ESC> ! h <DEL>

Status: . Success

14.105 SET DELIMITERS

The escape sequence includes 2 ASCII character codes specifying the new escape sequence initiator and terminator.

```
<ESC> <current_initiator> i <hi_new-initiator> <lo_new-initiator>
                            <hi_new-terminator> <lo_new-terminator>
                            <current_terminator>
```

Status: . Success
P Invalid character codes

Notes: o See the ALIAS command

14.106 SET STUTTER

The escape sequence includes an ASCII 0 or 1 to disable or enable the "stutter" delay from arrival of <terminator> until the first byte of the reply string (data or status) is sent to the host.

<ESC> ! j <ascii_0-or-1> <DEL>

Status: ? Invalid command

Notes: o 10 msec < stutter < 20 msec
o Characters other than the first are sent at the normal rate

14.107 SHOW STUTTER

The Q-Monitor returns an ASCII 0 or 1, as defined in SET STUTTER.

<ESC> ! k <DEL>

Status: N/A

14.108 SHOW BUILD

The Q-Monitor returns its PROM's build date and time as "dd-mmm-yy hh:mm:ss".
<ESC> ! l ... <DEL> (the lower case letter L)

Status: N/A

14.109 Reserved

<ESC> ! m ... <DEL>

Status: ? Invalid command

14.110 Reserved

<ESC> ! n ... <DEL>

Status: ? Invalid command

14.111 Reserved

<ESC> ! o ... <DEL>

Status: ? Invalid command

14.112 Reserved

<ESC> ! p ... <DEL>

Status: ? Invalid command

14.113 Reserved

<ESC> ! q ... <DEL>

Status: ? Invalid command

14.114 Reserved

<ESC> ! r ... <DEL>

Status: ? Invalid command

14.115 Reserved

<ESC> ! s ... <DEL>

Status: ? Invalid command

14.116 Reserved

<ESC> ! t ... <DEL>

Status: ? Invalid command

14.117 Reserved

<ESC> ! u ... <DEL>

Status: ? Invalid command

14.118 Reserved

<ESC> ! v ... <DEL>

Status: ? Invalid command

14.119 Reserved

<ESC> ! w ... <DEL>

Status: ? Invalid command

14.120 Reserved

<ESC> ! x ... <DEL>

Status: ? Invalid command

14.121 Reserved

<ESC> ! y ... <DEL>

Status: ? Invalid command

14.122 Reserved

<ESC> ! z ... <DEL>

Status: ? Invalid command

14.123 Reserved

<ESC> ! { ... <DEL>

Status: ? Invalid command

14.124 Reserved

<ESC> ! | ... <DEL>

Status: ? Invalid command

14.125 Reserved

<ESC> ! } ... <DEL>

Status: ? Invalid command

14.126 Reserved

<ESC> ! ~ ... <DEL>

Status: ? Invalid command

14.127 The ALIAS Command

This escape sequence is used in place of the command which is displaced if you specify an alternate escape sequence <terminator>. For example, if you change <terminator> to the ASCII character A (decimal code 65), you cannot issue the SHOW BAUD command (<ESC> <initiator> A <terminator>); however, if you use the "ALIAS" sequence

<ESC> <initiator> <DEL> <terminator>

the Q-Monitor will treat it as if it were a SHOW BAUD command.

The format of the ALIAS command, and the status or data returned by it, are the same as those of the command for which it substitutes, except that the normal command code (the 2nd character after the <ESC>) is replaced by <DEL>.

Notes:
- The Q-Monitor makes the association automatically between the ALIAS command and the "usurped" command code whenever you issue the SET DELIMITERS command
- The ALIAS command is ignored if <terminator> = <DEL>

BLANK

APPENDIX A

Command Summaries

NOTE

- Commands available in QMsetup are indicated by "*".
- Decimal command codes correspond to section numbers in "Command Descriptions" chapter.

A.1 Commands Ordered by ASCII Command Code

<ESC> <initiator> ...	Command Code Octal	Decimal	Hex	Mnemonic
<NUL>...<terminator>	0	0	0	Reserved
<SOH>...<terminator>	1	1	1	Reserved
<STX>...<terminator>	2	2	2	Reserved
<ETX>...<terminator>	3	3	3	Reserved
<EOT>...<terminator>	4	4	4	Reserved
<ENQ>...<terminator>	5	5	5	Reserved
<ACK>...<terminator>	6	6	6	Reserved
<BEL>...<terminator>	7	7	7	Reserved
<BS>...<terminator>	10	8	8	Reserved
<HT>...<terminator>	11	9	9	Reserved
<LF>...<terminator>	12	10	A	Reserved
<VT>...<terminator>	13	11	B	Reserved
<FF>...<terminator>	14	12	C	Reserved
<CR>...<terminator>	15	13	D	Reserved
<SO>...<terminator>	16	14	E	Reserved
<SI>...<terminator>	17	15	F	Reserved
<DLE>...<terminator>	20	16	10	Reserved
<DC1>...<terminator>	21	17	11	Reserved
<DC2>...<terminator>	22	18	12	Reserved
<DC3>...<terminator>	23	19	13	Reserved
<DC4>...<terminator>	24	20	14	Reserved
<NAK>...<terminator>	25	21	15	Reserved
<SYN>...<terminator>	26	22	16	Reserved
<ETB>...<terminator>	27	23	17	Reserved
<CAN>...<terminator>	30	24	18	Reserved
<EM>...<terminator>	31	25	19	Reserved
<SUB>...<terminator>	32	26	1A	Reserved
<ESC>...<terminator>	33	27	1B	Reserved
<FS>...<terminator>	34	28	1C	Reserved
<GS>...<terminator>	35	29	1D	Reserved
<RS>...<terminator>	36	30	1E	Reserved
<US>...<terminator>	37	31	1F	Reserved
<SP>...<terminator>	40	32	20	SET CLOCK *
! <terminator>	41	33	21	SHOW CLOCK *
" ... <terminator>	42	34	22	SET WATCHDOGS
# <terminator>	43	35	23	SHOW WATCHDOGS
$ ... <terminator>	44	36	24	SET ACTIONS
% <terminator>	45	37	25	SHOW ACTIONS
& ... <terminator>	46	38	26	SET REFERENCE
' <terminator>	47	39	27	SHOW REFERENCE

<ESC> <initiator> ...	Command Code Octal	Decimal	Hex	Mnemonic	
(... <terminator>	50	40	28	SET BATTERY	
) <terminator>	51	41	29	SHOW BATTERY	
* <terminator>	52	42	2A	SHOW DOWNTIME	
+ <terminator>	53	43	2B	RECALL FACTORY	*
, <terminator>	54	44	2C	SAVE	*
- <terminator>	55	45	2D	RECALL SAVED	*
. ... <terminator>	56	46	2E	SET VECTORS	*
/ <terminator>	57	47	2F	SHOW VECTORS	*
0 ... <terminator>	60	48	30	SET PRIORITIES	*
1 <terminator>	61	49	31	SHOW PRIORITIES	*
2 ... <terminator>	62	50	32	SET GREETING	
3 <terminator>	63	51	33	SHOW GREETING	
4 ... <terminator>	64	52	34	SET PARITY	*
5 <terminator>	65	53	35	SHOW PARITY	*
6 ... <terminator>	66	54	36	SET TABLE	
7 <terminator>	67	55	37	SHOW TABLE	
8 ... <terminator>	70	56	38	RESET LOG	
9 ... <terminator>	71	57	39	SHOW RECORD	
: ... <terminator>	72	58	3A	SET MASK	
; <terminator>	73	59	3B	SHOW MASK	
< ... <terminator>	74	60	3C	SET CONVERTER PARAMETERS	
= <terminator>	75	61	3D	SHOW CONVERTER PARAMETERS	
> <terminator>	76	62	3E	SHOW CONVERTER DATA	
? <terminator>	77	63	3F	RESET CONVERTER DATA	
@ ... <terminator>	100	64	40	SET BAUD	*
A <terminator>	101	65	41	SHOW BAUD	*
B <terminator>	102	66	42	SHOW UPTIME	
C ... <terminator>	103	67	43	SET LIMIT	
D <terminator>	104	68	44	SHOW LIMIT	
E ... <terminator>	105	69	45	SET BREAK	
F <terminator>	106	70	46	SHOW BREAK	
G ... <terminator>	107	71	47	SET PORT DATA	
H <terminator>	110	72	48	SHOW PORT DATA	
I ... <terminator>	111	73	49	SET PORT CONFIGURATION	
J <terminator>	112	74	4A	SHOW PORT CONFIGURATION	
K ... <terminator>	113	75	4B	SET TIMER CONSTANT	
L ... <terminator>	114	76	4C	SHOW TIMER CONSTANT	
M ... <terminator>	115	77	4D	SHOW TIMER COUNT	
N ... <terminator>	116	78	4E	SET TIMER MODE	
O ... <terminator>	117	79	4F	SHOW TIMER MODE	

<ESC> <initiator> ...	Command Code Octal	Decimal	Hex	Mnemonic	
P ... <terminator>	120	80	50	SET TIMER LINK	
Q <terminator>	121	81	51	SHOW TIMER LINK	
R ... <terminator>	122	82	52	SET TIMER GCB	
S ... <terminator>	123	83	53	SET TIMER TCB	
T ... <terminator>	124	84	54	SHOW TIMER CSR	
U ... <terminator>	125	85	55	SET UPS CONFIGURATION	
V <terminator>	126	86	56	SHOW UPS CONFIGURATION	
W <terminator>	127	87	57	BOOT	
X ... <terminator>	130	88	58	SET ALARM	
Y ... <terminator>	131	89	59	SET FAILOVER OUT	
Z ... <terminator>	132	90	5A	SET FAILOVER CONFIGURATION	
[<terminator>	133	91	5B	SHOW FAILOVER CONFIGURATION	
\ <terminator>	134	92	5C	SHOW UPS INPUT	
] <terminator>	135	93	5D	SHOW FAILOVER INPUT	
^ ... <terminator>	136	94	5E	SET PORT PROTECTION	
_ <terminator>	137	95	5F	SHOW PORT PROTECTION	
` ... <terminator>	140	96	60	SET PORT BUFFER	*
a <terminator>	141	97	61	SHOW PORT BUFFER	*
b <terminator>	142	98	62	SHOW INTERRUPT	
c ... <terminator>	143	99	63	SET TERMINATOR	*
d <terminator>	144	100	64	SHOW TERMINATOR	*
e <terminator>	145	101	65	SHOW ID	*
f ... <terminator>	146	102	66	SET DELAY	
g <terminator>	147	103	67	SHOW DELAY	
h <terminator>	150	104	68	RESET WATCHDOG	
i ... <terminator>	151	105	69	SET DELIMITERS	
j ... <terminator>	152	106	6A	SET STUTTER	
k <terminator>	153	107	6B	SHOW STUTTER	
l ... <terminator>	154	108	6C	SHOW BUILD	
m ... <terminator>	155	109	6D	Reserved	
n ... <terminator>	156	110	6E	Reserved	
o ... <terminator>	157	111	6F	Reserved	
p ... <terminator>	160	112	70	Reserved	
q ... <terminator>	161	113	71	Reserved	
r ... <terminator>	162	114	72	Reserved	
s ... <terminator>	163	115	73	Reserved	
t ... <terminator>	164	116	74	Reserved	
u ... <terminator>	165	117	75	Reserved	
v ... <terminator>	166	118	76	Reserved	
w ... <terminator>	167	119	77	Reserved	

<ESC> <initiator> ...	Command Code Octal/Decimal/Hex			Mnemonic
x ... <terminator>	170	120	78	Reserved
y ... <terminator>	171	121	79	Reserved
z ... <terminator>	172	122	7A	Reserved
{ ... <terminator>	173	123	7B	Reserved
| ... <terminator>	174	124	7C	Reserved
} ... <terminator>	175	125	7D	Reserved
~ ... <terminator>	176	126	7E	Reserved
<DEL>...<terminator>	177	127	7F	ALIAS

A.2 Commands Ordered Alphabetically by Mnemonic

Mnemonic	<ESC> <initiator> ...	Command Code Octal/Decimal/Hex		
ALIAS	<DEL>...<terminator>	177	127	7F
BOOT	W <terminator>	127	87	57
* RECALL FACTORY	+ <terminator>	53	43	2B
* RECALL SAVED	- <terminator>	55	45	2D
RESET CONVERTER DATA	? <terminator>	77	63	3F
RESET LOG	8 ... <terminator>	70	56	38
RESET WATCHDOG	h <terminator>	150	104	68
* SAVE	, <terminator>	54	44	2C
SET ACTIONS	$ ... <terminator>	44	36	24
SET ALARM	X ... <terminator>	130	88	58
SET BATTERY	(... <terminator>	50	40	28
* SET BAUD	@ ... <terminator>	100	64	40
SET BREAK	E ... <terminator>	105	69	45
* SET CLOCK	<SP>...<terminator>	40	32	20
SET CONVERTER PARAMETERS	< ... <terminator>	74	60	3C
SET DELAY	f ... <terminator>	146	102	66
SET DELIMITERS	i ... <terminator>	151	105	69
SET FAILOVER CONFIGURATION	Z ... <terminator>	132	90	5A
SET FAILOVER OUT	Y ... <terminator>	131	89	59
SET GREETING	2 ... <terminator>	62	50	32
SET LIMIT	C ... <terminator>	103	67	43
SET MASK	: ... <terminator>	72	58	3A
* SET PARITY	4 ... <terminator>	64	52	34
* SET PORT BUFFER	` ... <terminator>	140	96	60
SET PORT CONFIGURATION	I ... <terminator>	111	73	49
SET PORT DATA	G ... <terminator>	107	71	47
SET PORT PROTECTION	^ ... <terminator>	136	94	5E
* SET PRIORITIES	0 ... <terminator>	60	48	30
SET REFERENCE	& ... <terminator>	46	38	26
SET STUTTER	j ... <terminator>	152	106	6A
SET TABLE	6 ... <terminator>	66	54	36
* SET TERMINATOR	c ... <terminator>	143	99	63

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
SET TIMER CONSTANT	K ... <terminator>	113	75	4B
SET TIMER GCB	R ... <terminator>	122	82	52
SET TIMER LINK	P ... <terminator>	120	80	50
SET TIMER MODE	N ... <terminator>	116	78	4E
SET TIMER TCB	S ... <terminator>	123	83	53
SET UPS CONFIGURATION	U ... <terminator>	125	85	55
* SET VECTORS	. ... <terminator>	56	46	2E
SET WATCHDOGS	" ... <terminator>	42	34	22
SHOW ACTIONS	% <terminator>	45	37	25
SHOW BATTERY	) <terminator>	51	41	29
* SHOW BAUD	A <terminator>	101	65	41
SHOW BREAK	F <terminator>	106	70	46
SHOW BUILD	l <terminator>	154	108	6C
* SHOW CLOCK	! <terminator>	41	33	21
SHOW CONVERTER DATA	> <terminator>	76	62	3E
SHOW CONVERTER PARAMETERS	= <terminator>	75	61	3D
SHOW DELAY	g <terminator>	147	103	67
SHOW DOWNTIME	* <terminator>	52	42	2A
SHOW FAILOVER CONFIGURATION	[<terminator>	133	91	5B
SHOW FAILOVER INPUT	] <terminator>	135	93	5D
SHOW GREETING	3 <terminator>	63	51	33
* SHOW ID	e <terminator>	145	101	65
SHOW INTERRUPT	b <terminator>	142	98	62
SHOW LIMIT	D <terminator>	104	68	44
SHOW MASK	; <terminator>	73	59	3B
* SHOW PARITY	5 <terminator>	65	53	35
* SHOW PORT BUFFER	a <terminator>	141	97	61
SHOW PORT CONFIGURATION	J <terminator>	112	74	4A
SHOW PORT DATA	H <terminator>	110	72	48
SHOW PORT PROTECTION	_ <terminator>	137	95	5F
* SHOW PRIORITIES	~ <terminator>	61	49	31
SHOW RECORD	9 ... <terminator>	71	57	39
SHOW REFERENCE	' <terminator>	47	39	27
SHOW STUTTER	k <terminator>	153	107	6B
SHOW TABLE	7 <terminator>	67	55	37

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
* SHOW TERMINATOR	d <terminator>	144	100	64
SHOW TIMER CONSTANT	L ... <terminator>	114	76	4C
SHOW TIMER COUNT	M ... <terminator>	115	77	4D
SHOW TIMER CSR	T ... <terminator>	124	84	54
SHOW TIMER LINK	Q <terminator>	121	81	51
SHOW TIMER MODE	O ... <terminator>	117	79	4F
SHOW UPS CONFIGURATION	V <terminator>	126	86	56
SHOW UPS INPUT	\ <terminator>	134	92	5C
SHOW UPTIME	B <terminator>	102	66	42
* SHOW VECTORS	/ <terminator>	57	47	2F
SHOW WATCHDOGS	# <terminator>	43	35	23

A.3 Commands Ordered by Functional Group

A.3.1 Startup Greeting

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
SET DELAY	f ... <terminator>	146	102	66
SET GREETING	2 ... <terminator>	62	50	32
SHOW DELAY	g <terminator>	147	103	67
SHOW GREETING	3 <terminator>	63	51	33

A.3.2 Configuration

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
* RECALL FACTORY	+ <terminator>	53	43	2B
* RECALL SAVED	- <terminator>	55	45	2D
* SAVE	, <terminator>	54	44	2C

A.3.3 Communications and Terminal Emulation

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
* SET BAUD	@ ... <terminator>	100	64	40
SET BREAK	E ... <terminator>	105	69	45
SET DELIMITERS	i ... <terminator>	151	105	69
SET LIMIT	C ... <terminator>	103	67	43
* SET PARITY	4 ... <terminator>	64	52	34
SET STUTTER	j ... <terminator>	152	106	6A
* SET TERMINATOR	c ... <terminator>	143	99	63
* SHOW BAUD	A <terminator>	101	65	41
SHOW BREAK	F <terminator>	106	70	46
SHOW LIMIT	D <terminator>	104	68	44
* SHOW PARITY	5 <terminator>	65	53	35
SHOW STUTTER	k <terminator>	153	107	6B
* SHOW TERMINATOR	d <terminator>	144	100	64

A.3.4 Timekeeping

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
SET BATTERY	(... <terminator>	50	40	28
* SET CLOCK	<SP>...<terminator>	40	32	20
SHOW BATTERY	) <terminator>	51	41	29
* SHOW CLOCK	! <terminator>	41	33	21
SHOW DOWNTIME	* <terminator>	52	42	2A
SHOW UPTIME	B <terminator>	102	66	42

A.3.5 Watchdog Timers

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
RESET WATCHDOG	h <terminator>	150	104	68
SET WATCHDOGS	" ... <terminator>	42	34	22
SHOW WATCHDOGS	# <terminator>	43	35	23

A.3.6 Counter/Timers

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
SET TIMER CONSTANT	K ... <terminator>	113	75	4B
SET TIMER GCB	R ... <terminator>	122	82	52
SET TIMER LINK	P ... <terminator>	120	80	50
SET TIMER MODE	N ... <terminator>	116	78	4E
SET TIMER TCB	S ... <terminator>	123	83	53
SHOW TIMER CONSTANT	L ... <terminator>	114	76	4C
SHOW TIMER COUNT	M ... <terminator>	115	77	4D
SHOW TIMER CSR	T ... <terminator>	124	84	54
SHOW TIMER LINK	Q <terminator>	121	81	51
SHOW TIMER MODE	O ... <terminator>	117	79	4F

A.3.7 A/D Converter

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
RESET CONVERTER DATA	? <terminator>	77	63	3F
SET CONVERTER PARAMETERS	< ... <terminator>	74	60	3C
SHOW CONVERTER DATA	> <terminator>	76	62	3E
SHOW CONVERTER PARAMETERS	= <terminator>	75	61	3D

A.3.8 Digital I/O Port

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
* SET PORT BUFFER	` ... <terminator>	140	96	60
SET PORT CONFIGURATION	I ... <terminator>	111	73	49
SET PORT DATA	G ... <terminator>	107	71	47
SET PORT PROTECTION	^ ... <terminator>	136	94	5E
* SHOW PORT BUFFER	a <terminator>	141	97	61
SHOW PORT CONFIGURATION	J <terminator>	112	74	4A
SHOW PORT DATA	H <terminator>	110	72	48
SHOW PORT PROTECTION	_ <terminator>	137	95	5F

A.3.9 Event Monitoring

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
RESET LOG	8 ... <terminator>	70	56	38
SET ACTIONS	$ ... <terminator>	44	36	24
SET REFERENCE	& ... <terminator>	46	38	26
SET TABLE	6 ... <terminator>	66	54	36
SHOW ACTIONS	% <terminator>	45	37	25
SHOW RECORD	9 ... <terminator>	71	57	39
SHOW REFERENCE	' <terminator>	47	39	27
SHOW TABLE	7 <terminator>	67	55	37

A.3.10 UPS and Failover

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
SET FAILOVER CONFIGURATION	Z ... <terminator>	132	90	5A
SET FAILOVER OUT	Y ... <terminator>	131	89	59
SET UPS CONFIGURATION	U ... <terminator>	125	85	55
SHOW FAILOVER CONFIGURATION	[<terminator>	133	91	5B
SHOW FAILOVER INPUT	] <terminator>	135	93	5D
SHOW UPS CONFIGURATION	V <terminator>	126	86	56
SHOW UPS INPUT	\ <terminator>	134	92	5C

A.3.11 Interrupt Control

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
SET MASK	: ... <terminator>	72	58	3A
* SET PRIORITIES	0 ... <terminator>	60	48	30
* SET VECTORS	. ... <terminator>	56	46	2E
SHOW INTERRUPT	b <terminator>	142	98	62
SHOW MASK	; <terminator>	73	59	3B
* SHOW PRIORITIES	1 <terminator>	61	49	31
* SHOW VECTORS	/ <terminator>	57	47	2F

A.3.12 Miscellaneous

Mnemonic	<ESC> <initiator> ...	Command Code Octal	Decimal	Hex
ALIAS	<DEL>...<terminator>	177	127	7F
BOOT	W <terminator>	127	87	57
SET ALARM	X ... <terminator>	130	88	58
SHOW BUILD	l <terminator>	154	108	6C
* SHOW ID	e <terminator>	145	101	65

APPENDIX B

Factory Defaults

B.1 Communications

Startup Delay	4 sec approx.		(SET DELAY 6)
BREAK recognition	Enabled		(SET BREAK 1)
100 cps limit	Enabled		(SET LIMIT 1)
"Stutter"	Enabled		(SET STUTTER 1)
Parity	Disabled		(SET PARITY D)
Baud rate	110		(SET BAUD 9)
Reply terminator	13	(<CR>)	(SET TERMINATOR 0D)
Escape sequence ...			
initiator	33	("!")	
terminator	127	(<DEL>)	(SET DELIMITERS 21 7F)

B.2 Interrupts

Interrupt source	Vector (octal)	Q-Bus priority	
Terminal I/O	330	4	(Including Q-Monitor commands and data)
Watchdog B	270	5	
UPS transitions	270	5	
A/D excursions	270	5	
Digital I/O	270	5	
CT1	270	5	
CT2	270	5	
Failover In	270	5	

All Event interrupts are disabled (as after SET MASK 0).

B.3 Watchdogs and Counter/Timers

Watchdog A	5 min	(constant = 3000 @ 10 Hz)
Watchdog B	5 min	(constant = 3000 @ 10 Hz)
Counter/Timer 1	Off	(constant = 65536, counter is inactive)
Counter/Timer 2	Off	(constant = 65536, counter is inactive)
Counter/Timer linkage	0	(counter/timers are independent)
Counter/Timer modes	0	(continuous pulse mode, retrigger disabled, no external access enabled)

B.4 Event Logging

Feature	Configuration
UPS status	Active High
Failover In	Active Low
Failover Out	Active Low

Event	Action	Records allocated	Logging mode
Warm starts	COUNT & LOG	10	Purge
Cold starts	COUNT & LOG	10	Purge
Watchdog A	COUNT & LOG	10	Purge
Watchdog B	COUNT & LOG	10	Purge
UPS transitions	NOP	10	Purge
A/D excursions	NOP	10	Purge
Digital I/O	NOP	10	Purge
Counter/Timer 1	COUNT & LOG	10	Purge
Counter/Timer 2	COUNT & LOG	10	Purge
Failover In	NOP	10	Purge

B.5 A/D Converter

Channel	Upper threshold	Lower threshold	
0	255	0	(redundant +12V supply)
1	255	0	(redundant +5V supply)
2	255	0	
3	255	0	(redundant temperature)
4	255	0	
5	255	0	Host +12V supply
6	255	0	Host +5V supply
7	255	0	Host temperature

B.6 Digital I/O

Buffer type 1 (SET PORT BUFFER 1)
Protection mask 2C (hex) (SET PORT PROTECTION 0 20 04)

Port bit	JB pin	I/O	Normal/ inverting	Pattern sensitivity	Function
0	1	output	normal	none	user
1	2	output	normal	none	user
2	3	output	invert	none	user or Failover Out
3	-	input	normal	0-to-1 transition	Q-Bus I/O
4	5	output	normal	none	user
5	6	input	normal	any transition	user or UPS input
6	7	input	normal	none	user
7	8	input	normal	none	user

Pattern Polarity = 28 (hex)
Pattern Transition = 28 (hex)
Pattern Mask = 28 (hex)

Data Path Polarity = 04

Data Direction = E8 (hex)
Special I/O = 00

B.7 Miscellaneous

Hobbs meters, reference date, battery installation date, and the "greeting" are all undefined.

BLANK

APPENDIX C

Support Kit

Each Q-Monitor is shipped with a Support Kit containing programs and information to expedite integration of the Q-Monitor into your system. Depending upon the operating system you are using, the contents of the Support Kit may vary but sources are always included for any programs in the Kit. Each Support Kit contains a file named README1ST (README.1ST for RT11) which describes the contents of the kit and the specific installation procedures.

Always included is QMCMDS, a package of Fortran subroutines which execute the complete set of Q-Monitor commands. QMCMDS routines all communicate with the Q-Monitor through a single subroutine named QMSEND. The Support Kit may include several different versions of QMSEND which use different access techniques (terminal driver, DEV partition, physical page mapping, etc.) to address the Q-Monitor.

For each supported operating system (currently the RSX variants, RT11 and MicroVMS), the Kit includes at least one version of a program called SETUP, and at least one version of a program called RTIME.

SETUP is a Fortran program which accepts the English equivalents of Q-Monitor commands (as listed in the Command Summaries) typed at a terminal, calls QMCMDS routines to issue the corresponding escape sequences to the Q-Monitor, and then displays on the terminal any data or status returned by the Q-Monitor. For each supported system, a version of SETUP is provided which assumes that the terminal used to type the commands IS NOT connected to the Q-Monitor, and that the terminal connected to the Q-Monitor IS NOT recognized by the system's terminal driver.

Additionly, for those systems whose terminal drivers support DLV11s, at least one version of SETUP is provided which assumes that the terminal used to type the commands IS connected to the Q-Monitor, and that it IS recognized by the the system's terminal driver.

RTIME calls the QMCMDS subroutine SHCLOC to issue a "SHOW CLOCK" command, and then uses the date and time returned to set the operating system's date and time. As for SETUP, a non-terminal driver version of RTIME is provided for each supported operating system, and terminal driver versions are provided for those systems that support the DLV11.

Other programs and information specific to your operating system may also be included.

BLANK

APPENDIX D

Example Fortran Program

The program listed here uses an ordinary WRITE (statement 1) to send the SHOW CLOCK command, then an ordinary READ (statement 2) to get the 16-character reply (the date and time). Since the Q-Monitor's default <reply_terminator> is the carriage return, no special action is needed in the READ.

Please note that statements 1 and 2 are the only ones in the program that have anything to do with the Q-Monitor; the remainder of the program is there simply to format and print the date and time as a text string (CHTB is a function which returns the binary value corresponding to an ASCII hexadecimal digit).

This program has been run on RT11 using the FORTRAN IV compiler, and on RSX11M using both FORTRAN IV and FORTRAN IV-PLUS. Other compilers and/or operating systems should present little challenge, but the program uses a few conveniences that are not ANSI compatible.

The only requirements are that:

- Your operating system's terminal driver must be configured to support a DLV11 interface with the CSR address and vector that your Q-Monitor uses.
- You must assign Fortran logical unit number 5 to the terminal driver unit corresponding to the Q-Monitor (the RT11 and RSX linkers automatically assign LUN 5 to the terminal from which the program runs).
- The Q-Monitor's LIMIT and STUTTER features should be enabled; the program may work with these disabled, but it will depend upon the speed of the processor, operating system, and compiled code.
- The escape sequence delimiters and the reply terminator must be the factory standards.

Specifically, this program does not require:

- access to the I/O page,
- any operating system specific features,
- privileged task status, or
- linking with anything other than the Fortran runtime library.

```
      PROGRAM FUBAR
      BYTE CHTB, IDT(8), COMAND(4), STRING(80)
      REAL*4 DAYS(7), DAY, MONS(12), MON
      DATA DAYS /'MON ', 'TUE ', 'WED ', 'THU ', 'FRI ', 'SAT ', 'SUN '/
      DATA MONS /'JAN-', 'FEB-', 'MAR-', 'APR-', 'MAY-', 'JUN-',
```

```
     +          'JUL-', 'AUG-', 'SEP-', 'OCT-', 'NOV-', 'DEC-'/

      DATA COMAND /27, '[', '[', 127/          ! <ESC> ! ! <DEL>

1     WRITE (5,5001) COMAND                    ! Send SHOW CLOCK command
2     READ (5,5000) (STRING(I), I=1,16)        ! Read 16 hex characters
      DO 10 I = 1,8                            ! Convert 16 hex digits
       IDT(I) = 16 * CHTB(STRING(2*I-1))       !  to 8 bytes
     +           + CHTB(STRING(2*I))           !
10    CONTINUE                                 !
      DAY = DAYS(IDT(8))                       ! Index the day-of-week table
      MON = MONS(IDT(2))                       ! Index the month table
      WRITE (5,5003) DAY, IDT(3), MON,         ! Print "ddd dd-mmm-
     +               IDT(1), IDT(4), IDT(5),   !       yy hh:mm:
     +               IDT(6), IDT(7)            !       ss.t"
      STOP                                     ! (may be some missing zeros)

5000  FORMAT (16A1)
5001  FORMAT ('$',4A1)
5003  FORMAT (1X,A4,I2'-'A4,I2' 'I2':'I2':'I2'.'I1)
      END

      BYTE FUNCTION CHTB (H)
      BYTE H

!
! "BYTE-valued" function to convert an ASCII hexadecimal character to binary
! (no error indication is returned for invalid characters)
!

      CHTB = 0                          ! Return 0 for invalid codes
      IF (H .LT. '0') RETURN            ! If LT invalid
      IF (H .GT. '9') GO TO 10          ! If GT maybe A-F
      CHTB = H - 48                     ! 0-9 -- ditch the ASCII bits
      RETURN                            !
10    IF (H .LT. 'A') RETURN            ! If LT invalid
      IF (H .GT. 'F') RETURN            ! If GT invalid
      CHTB = H - 55                     ! A-F -- convert to 10-15
      RETURN

      END
```

APPENDIX E

7-Bit ASCII Codes

CHAR	OCT	DEC	HEX	CHAR	OCT	DEC	HEX	CHAR	OCT	DEC	HEX	CHAR	OCT	DEC	HEX
NUL	000	000	00	SP	040	032	20	@	100	064	40	`	140	096	60
SOH	001	001	01	!	041	033	21	A	101	065	41	a	141	097	61
STX	002	002	02	"	042	034	22	B	102	066	42	b	142	098	62
ETX	003	003	03	#	043	035	23	C	103	067	43	c	143	099	63
EOT	004	004	04	$	044	036	24	D	104	068	44	d	144	100	64
ENQ	005	005	05	%	045	037	25	E	105	069	45	e	145	101	65
ACK	006	006	06	&	046	038	26	F	106	070	46	f	146	102	66
BEL	007	007	07	'	047	039	27	G	107	071	47	g	147	103	67
BS	010	008	08	(	050	040	28	H	110	072	48	h	150	104	68
HT	011	009	09	)	051	041	29	I	111	073	49	i	151	105	69
LF	012	010	0A	*	052	042	2A	J	112	074	4A	j	152	106	6A
VT	013	011	0B	+	053	043	2B	K	113	075	4B	k	153	107	6B
FF	014	012	0C	,	054	044	2C	L	114	076	4C	l	154	108	6C
CR	015	013	0D	-	055	045	2D	M	115	077	4D	m	155	109	6D
SO	016	014	0E	.	056	046	2E	N	116	078	4E	n	156	110	6E
SI	017	015	0F	/	057	047	2F	O	117	079	4F	o	157	111	6F
DLE	020	016	10	0	060	048	30	P	120	080	50	p	160	112	70
DC1	021	017	11	1	061	049	31	Q	121	081	51	q	161	113	71
DC2	022	018	12	2	062	050	32	R	122	082	52	r	162	114	72
DC3	023	019	13	3	063	051	33	S	123	083	53	s	163	115	73
DC4	024	020	14	4	064	052	34	T	124	084	54	t	164	116	74
NAK	025	021	15	5	065	053	35	U	125	085	55	u	165	117	75
SYN	026	022	16	6	066	054	36	V	126	086	56	v	166	118	76
ETB	027	023	17	7	067	055	37	W	127	087	57	w	167	119	77
CAN	030	024	18	8	070	056	38	X	130	088	58	x	170	120	78
EM	031	025	19	9	071	057	39	Y	131	089	59	y	171	121	79
SUB	032	026	1A	:	072	058	3A	Z	132	090	5A	z	172	122	7A
ESC	033	027	1B	;	073	059	3B	[	133	091	5B	{	173	123	7B
FS	034	028	1C	<	074	060	3C	\	134	092	5C	|	174	124	7C
GS	035	029	1D	=	075	061	3D	]	135	093	5D	}	175	125	7D
RS	036	030	1E	>	076	062	3E	^	136	094	5E	~	176	126	7E
US	037	031	1F	?	077	063	3F	_	137	095	5F	DEL	177	127	7F

BLANK

APPENDIX F
INDEX II

100 cps limit 1-5, 10-2, 14-20
12 volt supply 1-4
12V supply 6-1

5 volt supply 1-4
5V supply 6-1

<initiator> 13-1, 14-1
<reply_terminator> 13-1, 13-3, 14-1, 14-14, 14-29
<terminator> 13-1, 14-1, 14-34

A/D channels 1-4, 6-1
A/D conversion factors 6-2
A/D converter 1-4, 6-1, 14-17, 14-18
A/D excursions 1-4, 14-16
A/D thresholds 1-4, 14-18
Action codes 1-3, 2-2, 2-5, 12-4, 14-8
Actions 1-2, 2-1, 4-1, 5-2, 14-8
Address PAL 1-5, 10-1

Battery usage, see Hobbs meter 0-1
Baud rate 1-5, 10-2, 14-19
BEVNT clock 5-2
BOOT 14-26
BREAK 1-5, 1-6, 11-1, 13-2, 14-20
Buffer chip 1-4, 7-1, 7-3, 14-21, 14-28

Calendar clock 1-3, 3-1, 8-1, 12-2
Commands 1-2, 1-6, 10-1, 11-1, 13-1, 14-1
Console ODT 10-2, 11-1
Console terminal 10-2, 11-1
Counter/timer 1-3, 5-1, 14-22, 14-23, 14-24, 14-25

Date and time 1-3, 2-5, 3-1, 8-1, 12-2, 14-6, 14-9, 14-10
Day of week 1-3, 3-1, 14-6, 14-9, 14-10
Defaults 1-5, 2-5, 7-1, 10-1, 12-1, 12-3, 13-2
Digital I/O 1-4, 5-1, 7-1, 7-3, 14-16
DLV11 . . . . . 1-5, 10-1
Downtime, see Hobbs meter 0-1

Escape sequences 1-2, 10-1, 13-1, 14-30
Event codes 2-3, 14-16
Event counter 1-2, 2-2, 2-5
Event logging 1-2, 2-2, 2-3, 2-5, 12-3, 14-15, 14-16
Event monitoring 1-2, 2-1
Event Table 1-2, 2-2, 11-2, 12-1, 12-3
Events 1-2, 2-1, 14-15, 14-16

Failover 2-1, 2-3, 7-1, 14-16, 14-21, 14-27, 14-28

Hobbs meter 1-4, 8-1, 14-10, 14-19

Keyboard commands, see QMsetup 0-1

Line time clock, see BEVNT clock 0-1
Logging mode 2-2, 2-5, 2-6, 14-15

Non-volatile RAM 1-4, 9-1, 12-2, 14-10

Parameter Page 1-2, 11-2, 12-1, 12-3, 14-11
Parity 1-5, 14-14
Pattern recognition 1-4, 7-2, 14-21

Q-Bus interrupts 1-2, 2-3, 2-6, 4-1, 10-1, 10-3, 11-1, 12-2, 14-17, 14-29
Q-bus priority 1-5, 14-13
Q-Bus registers 1-5, 2-4, 2-6, 10-1, 12-2, 13-2
Q-Bus vectors 1-5, 2-6, 10-1, 14-12
Q-Monitor status 10-1, 13-1
QMdebug 1-6, 11-2, 14-19
QMsetup 1-2, 1-6, 11-2, 13-1, 14-19

RBUF 2-6, 10-1, 10-3, 12-2
RCSR 1-3, 1-5, 10-1
RECALL FACTORY 14-11
RECALL SAVED 14-11
Registers 4-1
RESET CONVERTER DATA 6-2, 14-18
RESET LOG 2-2, 14-16
RESET WATCHDOG 1-3, 2-4, 4-1, 14-30
Resetting the watchdog 1-3, 4-1, 10-1, 13-2, 14-30
Restrictions 2-3, 7-1, 10-3, 11-1, 12-4
Resynchronization 13-2

SAVE 12-3, 14-11
Serial interface 1-2, 1-5, 10-1
SET ACTIONS 2-2, 14-8
SET ALARM 14-26
SET BATTERY 1-4, 8-1, 14-10
SET BAUD 10-2, 14-19
SET BREAK 11-1, 11-2, 14-20
SET CLOCK 1-3, 3-1, 14-6
SET CONVERTER PARAMETERS 6-2, 14-17

SET DELAY 14-30
SET DELIMITERS 14-1, 14-30
SET FAILOVER CONFIGURATION 14-21, 14-27
SET FAILOVER OUT 14-27
SET GREETING 14-13
SET LIMIT 10-2, 13-3, 14-20
SET MASK 14-17
SET PARITY 14-14
SET PORT BUFFER 5-2, 7-1, 7-3, 14-21, 14-28
SET PORT CONFIGURATION 5-2, 7-1, 7-2, 7-3, 14-21, 14-28
SET PORT DATA 7-3, 14-21, 14-28
SET PORT PROTECTION 7-3, 14-28
SET PRIORITIES 14-13
SET REFERENCE 14-9
SET STUTTER 10-2, 14-31
SET TABLE 2-2, 2-5, 14-15
SET TERMINATOR 13-1, 13-3, 14-29
SET TIMER CONSTANT 5-1, 5-2, 14-22, 14-25
SET TIMER GCB 5-1, 14-25
SET TIMER LINK 5-1, 5-2, 14-24
SET TIMER MODE 5-1, 5-2, 14-23, 14-25
SET TIMER TCB 5-1, 5-2, 14-25
SET UPS CONFIGURATION 14-21, 14-26
SET VECTORS 10-1, 14-12
SET WATCHDOGS 4-1, 14-7
SHOW ACTIONS 2-2, 14-8
SHOW BATTERY 14-10
SHOW BAUD 14-19
SHOW BREAK 14-20
SHOW BUILD 14-31
SHOW CLOCK 1-3, 3-1, 14-6
SHOW CONVERTER DATA 6-2, 14-18
SHOW CONVERTER PARAMETERS 6-2, 14-16
SHOW DELAY 14-30
SHOW DOWNTIME 1-4, 8-1, 14-10
SHOW FAILOVER CONFIGURATION 14-27
SHOW FAILOVER INPUT 14-27
SHOW GREETING 14-14
SHOW ID 14-29
SHOW INTERRUPT 14-29
SHOW LIMIT 14-20
SHOW MASK 14-17
SHOW PARITY 14-14
SHOW PORT BUFFER 14-28
SHOW PORT CONFIGURATION 14-22
SHOW PORT DATA 14-21
SHOW PORT PROTECTION 14-28
SHOW PRIORITIES 14-13
SHOW RECORD 2-5, 2-7, 14-16
SHOW REFERENCE 14-9
SHOW STUTTER 14-31
SHOW TABLE 2-2, 2-5, 2-6, 14-15, 14-16
SHOW TERMINATOR 14-29

SHOW TIMER CONSTANT 14-22
SHOW TIMER COUNT 5-1, 14-23
SHOW TIMER CSR 5-1, 14-25
SHOW TIMER LINK 14-24
SHOW TIMER MODE 14-24
SHOW UPS CONFIGURATION 14-26
SHOW UPS INPUT 14-27
SHOW UPTIME 1-4, 8-1, 14-19
SHOW VECTORS 14-12
SHOW WATCHDOGS 14-7
Special Data 2-5, 2-7, 14-16
Startup delay 12-1, 14-30
Startup message 13-3, 14-13, 14-14
Support Kit 1-1, 1-5, 10-3, 10-4

TBUF 1-3, 2-4, 4-1, 10-1, 13-2
TCSR 10-1
Temperature 1-4, 6-1
Terminal emulation 1-2, 1-5, 1-6, 10-1, 11-1, 11-2, 12-1, 13-1
The ALIAS command 14-34

UPS 2-3, 7-1, 14-16, 14-21, 14-26, 14-27, 14-28
Uptime, see Hobbs meter 0-1

Watchdog timer 1-3, 2-3, 4-1, 13-2, 14-7, 14-30

BLANK

BLANK